Overload Control for a System in Time-Varying Environment

Ohad Perry and Ward Whitt

January 22, 2013

Abstract

In recent papers we considered how two large service systems that are primarily designed to operate independently, can help each other in face of unexpected overloads, due to a sudden change in the arrival rates. We proposed an overload control, which we named *fixed-queue-ratio with thresholds* (FQR-T), whose aim was to prevent any sharing of customers, i.e., sending customers from one class to be served in the other class' pool, during normal loads, and to initiate sharing automatically once a threshold is crossed, in which case the corresponding pool is considered overloaded. The goal is to keep the relation between the two queues fixed at a certain ratio, which is optimal in a deterministic "fluid" approximation, assuming a holding cost is incurred on the two queues. To avoid harmful sharing our control includes the *one-way sharing rule*, stipulating that sharing is allowed in only one direction at any time. In this paper we consider a more complex time-varying environment, in which the arrival rates and staffing levels are time dependent, so that the system may fluctuate between periods of various loads, with overloads possible in either direction. We show that FQR-T needs to be modified to account for these more complex settings, since it may be slow to react to the changing environment, and may even cause sever fluctuations once the arrival rates return to normal after an overload incident. Our new control, *FQR with activation-and-release thresholds* (FQR-ART) is designed to automatically respond to changes in the environment by initiating sharing in the right direction quickly, if that is needed, while avoiding harmful phenomenons, such as congestion collapse and severe oscillations during normal loads. A novel fluid approximation, described implicitly via an *ordinary-differential equation* (ODE) is developed, as well as an efficient algorithm to solve that ODE.

1 Introduction

Queueing systems typically experience time-varying loads and may also experience periods of congestions. Overload incidents are prevalent phenomenons in communication networks, healthcare systems, call centers, etc. In this paper we are motivated by a call-center application, but the insights and methods we develop can be employed in other queueing settings.

In a typical call center, under normal circumstances, the arrival rates vary by time of day in a predictable way, and the staffing responds to that anticipated pattern, typically with fixed staffing levels over short time periods, such as half hours; see [1] and [13] for background. However, it is often not possible to have the exact staffing needed to meet pre-specified service-level constraints at all times. More specifically, there are time periods during the day in which the number of agents may be higher or lower than the number needed to provide the desired service levels for the customers in terms of, e.g., waiting times in queue. This can occur for several reasons such as (i) encountering arrival rates that are higher than initially forecasted;

(ii) the inability to have the staffing change together with the arrival rates appropriately due to human-resource constraints; (iii) agents absenteeism, e.g., agents not showing up to work, or taking unreported breaks during their shift; (iv) system malfunction, e.g., computer failure which prevents some, or all, agents from providing service. Such discrepancies between arrival rates and staffing, whether anticipated or not, can cause overloads in the system for long time periods, whose consequence is serious degradation of the system performance in terms of waiting times in queue, customer abandonment, etc. In addition, overloads can occur due to inappropriate control.

In this paper we continue our research on how to exploit flexibility of queueing systems in some "optimal" manner in order to alleviate congestions due to having arrival rates being are higher than the system's service capacity. In particular, our aim is to design an automatic and easy-to-implement control, with the objective of *efficiently* routing customers between two different service pools in a time-varying environment encountering periods of overloads. Although having flexibility in stochastic queueing networks is generally perceived useful, we show that it sometimes needs to be exploited with caution in order to avoid bad performance, such as chaotic oscillations and congestion collapse; see §4 below and §1.2 for other examples in the literature.

The Context. In [25] we considered how two service systems, each consisting of a single many-agent service pool having a designated customer class, can help each other in face of an unexpected overload that is due to a sudden increase in the arrival rates. Since sharing of customers, i.e., sending customers from one class to be served in the other class pool, is potentially possible in the two directions, the model is known in the call-center literature as an X model; see Figure 1.

In the setting of [25], once an unexpected overload occurs, and only then, one of the two customer classes, the one deemed "more overloaded", should receive help from the other service pool in such a way that a specific ratio holds approximately between the two queues. That ratio is found by solving a deterministic "fluid" optimization problem, assuming a holding cost is incurred on both queues during overload incidents. There are two possible different target ratios - one for each direction of sharing - with $r_{i,j}$ denoting the fluid-optimal ratio when queue i should receive help from pool j, $i, j = 1, 2$. Here we treat those two ratios as given parameters, and refer to [25] for further details on how to compute them.

To achieve the objectives described above we suggested the FQR-T control together with *one-way sharing*. The fixed-queue ratio is the (fluid-optimal) ratio discussed above, depending on the direction of overload, and the thresholds are two positive numbers relating to the queues, whose purpose is to prevent sharing when neither pool is overloaded, and detect overloads quickly so that sharing can begin in the right direction once an overload incident begins. More specifically, once one of those activation thresholds is crossed, the system is considered to be overloaded, with the direction of the overload depending on the specific threshold that was crossed; we elaborate in §2 below. The one-way sharing rule stipulates that no class-1 customers are allowed to be routed to pool 2 at any time $t \geq 0$ if there are any class-2 customers in pool 1 at that time t, and similarly in the other direction.

In [26] we developed a fluid model to facilitate the analysis of the complex behavior of the X system under FQR-T during overloads, and performed a detailed rigorous analysis of that model in [27]. Finally, in [28] we proved that the fluid model arises as a many-server heavy-traffic fluid limit of properly-scaled sequence of stochastic X systems. Diffusion-limit refinements for that fluid limit were proved in [29].

1.1 Contribution: Control and Fluid Analysis in Time-Varying Environment

Building on fluid analysis, in this paper we show that FQR-T with one-way sharing is not adequate for time-varying environment, since it may be too slow to react to changing loads. To accelerate the response to switching overloads, we modify the one-way sharing rule by allowing class-i customers to be routed to pool j, provided that there are **no more** than $\tau_{j,i} > 0$ class-j customers in pool i, $i \neq j$, if sharing is desired. We refer to $\tau_{1,2}$ and $\tau_{2,1}$ as “release thresholds” (to be distinguished from the “activation thresholds” discussed above). We name our modified control *Fixed-Queue-Ratio with Activation and Release Thresholds*, abbreviated **FQR-ART**.

As was previously observed in numerous other settings (see §1.2 below), overload controls may cause severe degradation of system’s performance during normal loads if they are not executed with caution. In our case, if the activation thresholds in FQR-ART are not chosen carefully, a *congestion collapse* may occur, namely, the effective service rate is reduced, so that a normally-loaded system becomes overloaded. A congestion collapse result was also observed in [25] and analyzed in [26] and was due to inefficient sharing in both directions simultaneously. (Which is the main reason for having the one-way sharing rule.) However, unlike in those previous papers, here that phenomenon is due to chaotic oscillations of the service process that may occur when a system recovers from an overload period, and is due to the new release thresholds. We solve that problem by choosing larger activation thresholds; see §4 below.

Our contribution is threefold: (i) We demonstrate how and when it is beneficial (and harmful) to exploit the flexibility of the system. (ii) We design a control that automatically exploits that flexibility when this is beneficial, and prevents bad behaviors that can occur in the time-varying settings. (iii) We develop a novel fluid model to approximate the (untractable) stochastic system in time-varying environment. That fluid model is described implicitly as the solution of an *ordinary differential equation* (ODE), building on a stochastic *averaging principle* (AP); see §5.2 below. Finally, we design an efficient algorithm to solve that ODE.

Here is how the rest of the paper is organized. We conclude this section with a literature review. In §2 we describe the stochastic X model and the FQR-T and FQR-ART controls. Building on simple fluid considerations, In §§3 and 4 we demonstrate the need to modify FQR-T in order to adjust to the time-varying settings considered in this paper. Specifically, in §3 we show why release thresholds are essential, and in §4 we show that, unless precaution is taken, the release thresholds can cause congestion collapse when the system recovers from an overload. Specifically, a system which has enough potential service capacity becomes overloaded due to oscillatory behavior of the service process. To avoid that bad behavior, the activation thresholds need to be of a larger order of size than those suggested for the FQR-T control. Section §5 is dedicated to a development of the fluid approximation to the system which is represented implicitly as a solution to an ODE. In §6 we explain how this ODE can be solved numerically and provide an efficient algorithm to numerically solve that ODE. In §7 we provide numerical examples, demonstrating the effectiveness of the fluid model by comparing the solution to the ODE’s of the various cases to simulation experiments. In §8 we discuss stationarity and quasistationarity of the fluid model. Finally, we conclude in §9 and suggest some possible extensions and further research.

1.2 Literature Review

Steady-state performance measures are sometimes possible to compute using, for example, standard CTMC theory or transforms methods; see, e.g., [9] and [31] for "exact analysis" of overload controls. If Laplace transforms are employed, then numerical algorithms to invert those transforms are needed; see, e.g., [4] for applications in related settings. However, as the models become more complicated, even numerical computations of steady state quantities and transforms become impractical. (For example, if the models are non-stationary; of high dimension; do not posses any form of reversibility; etc.) In particular, when overload controls are considered, it is important to analyze the time-dependent (non-stationary) evolution of the system under consideration in order to study its behavior when overloads begin as well as its recovery from overload incidents. Thus, even in the simplest Markovian settings, there is need to use approximations and/or simulations. Since we are concerned with the untractable behavior of a highly complex and non-stationary system, the following review focuses on examples of approximation methods and simulations to analyze complex queueing systems that are related, at least in some aspects, to our current study.

Time-Varying Systems. There is growing interest in systems having time-dependent arrival rates and staffing functions. For such systems, fluid models are especially valuable from both the operational and analytical perspectives. From the operational aspect, fluid models are important since they approximate the main trends in the system's behavior. Thus, from a managerial point of view, fluid models are superior to diffusion approximations in non-stationary settings. From the analytical aspect, fluid models are easier to derive (even if it is hard to prove that they arise as bona-fide weak limits) and are much more tractable than approximating diffusions; see [21, 22, 23] for current results on queues in time-varying environments experiencing periods of overloads, as well as for many more references on that subject. We do point out that to stabilize the system at a stationary state when it is experiencing time-varying arrival rates, we employ time-varying staffing functions based on the infinite-server approximation in [17]. See also §3 in [11]. (However, here we are concerned with fluid models, so we do not employ square-root staffing refinements, as in the latter citations, which have no effect on the fluid approximations.)

Routing, Scheduling and Staffing of Overloaded Systems. As staffing decisions for call centers are done in advance, it is often the case that the number of agents present cannot be changed in response to the realized arrival rates. A multiple class and pool call center model is studied in [16], assuming that the arrival rates to the system are themselves stochastic processes (e.g., the arrival process might be a doubly-stochastic Poisson process). Since fluid approximations are employed to optimize the system, with the objective of minimizing abandonment in addition to staffing costs, the analysis in [16] is carried out for congested systems.

In [5] a system with multiple streams of arriving jobs evolving in discrete time is considered during a temporary overload period, where the overload in that reference is defined via rate stability (there is no abandonment). The authors in [5] suggest using the max-weight policy which, much like the FQR-ART control, is simple to implement since it uses only information on the current state of the system; it stabilizes the system during normal-loads periods; and, finally, it keeps the queues at target ratios when the system cannot be stabilized due to high arrival rates.

It is significant that the max-weight policy is not order optimal for the X model, i.e., the steady state queues grow faster than $O(\sqrt{n})$ in the many-server heavy-traffic limiting regime, as shown in [34]. (A similar phenomenon was observed in [25], when FQR-T is employed with activation thresholds that are too small.) The authors in [34] propose a control which they name *shadow routing*, and prove diffusion limits under a complete-resource-pooling condition for a general parallel service system. Modifications of the shadow-routing control, designed for overloaded parallel systems with unknown arrival rates, are studied in a subsequent paper [35]. Similar to our case, the objective of the control is to quickly and automatically detect overloads, and route customers in an "optimal" manner. More precisely, asymptotic optimality of their SHADOW-RM control is proved, where the objective is to maximize the reward rate, assuming a class-dependent reward of each customer served.

In [15] overflow networks in heavy-traffic are studied in settings of co-sourcing, i.e., when firms that operate their own in-house call center overflow a *nonnegligible proportion* of the arrivals to a call center that is operated by an outsourcer. Specifically, the in-house service pool has a finite buffer, or no buffer at all, for waiting customers (in the latter case, the in-house pool is a loss system), and customers who arrive to find the system full are overflowed to the outsourcer. In [15], "nonnegligible proportion" of overflow implies that the arrival rate to the in-house pool is larger than its maximum service capacity, so that the in-house pool is overloaded in the sense that all agents are busy all the time, asymptotically, in the many-server heavy-traffic limit. Fluid and diffusion limits are obtained via a stochastic AP, and an asymptotic-independence result, which facilitates solving related optimization problems, is shown to hold.

Healthcare Systems. System Overloads are especially prevalent in healthcare systems, and can even be considered a "natural state" in many cases. For example, operation rooms, *intensive care units* (ICU), MRI machines, etc., are designed to operate continuously, 24/7, and exhibit long lines of waiting patients whose wait times can be in the order of weeks, months and even years.

A hospital, as a holistic system, has extremely complex queueing dynamics, having multiple internal flows between its units in addition to exogenous arrival streams. Thus, overloads in some units of a hospital can "propagate" to other units, creating a system-wide overload. For example, when *inpatient wards* (IW) are overloaded, patients from the *emergency department* (ED) who need to be hospitalized cannot be transferred to the IW due to the unavailability of beds, creating a well-known phenomenon of *blocked beds* in the ED, i.e., beds that are occupied by patients who finished their treatment in the ED; see, e.g., [3] for a current review. We further refer to [2] for a data-based study of queueing aspects in hospital settings, as well as for an extensive literature review.

In [6], a fluid approximation of an ICU experiencing periods of overload periods is studied, in which the service rate of current ICU patients increases (is "sped-up") if the number of patients that are waiting to be admitted to the ICU exceeds a certain threshold. In turn, the sped-up patients have an increased probability of readmission to the ICU, so that alleviating overloads by employing speedup increases future overloads.

The fluid model in [6] builds on an AP as an engineering principle, in the spirit of our work [26].

Based on an extensive empirical study of a Singaporean hospital, a complex stochastic network model is designed and simulated in [33], in order to approximate the inpatient operations in that hospital. During periods of congestion with large waiting times, ED patients may be sent to a non-designated IW if a wait-time threshold is crossed. That threshold changes dynamically with the time of the day. In the stochastic-network terminology, that overflow procedure accounts for an activation of a non-basic activity. In our model, a non-basic activity relates to the possibility of routing customers from one class to its non-designated service pool, and such an activity is active when the appropriate activation threshold is crossed.

In [30] a fluid model is used to study the effects that IW discharge timing has on the ED boarding times. Based on that model, the authors conclude that shifting the IW discharge time, to take place before the ED experiences its peak demand, can eliminate the overload in the ED. This conclusion, however, is not supported by the findings of [33].

Congestion Collapse and Oscillations. In our model, the more overloaded the system is, the more agents are working with shared customers, so that throughput due to service decreases if the service rates of shared customers is lower than that of the designated customers. Moreover, as we showed in [25] and [26], even an underloaded system can go through "congestion collapse", creating severe overloads when an inappropriate control is employed. Congestion-collapse phenomenons are also observed in [19] and [32], where increased loads may cause decreasing throughput rates, if an inadequate service discipline is employed.

Another feature of overload controls is that they might cause oscillations during normal loads. See §4 below for oscillatory behavior that may occur in our model if FQR-ART is employed inadequately. Other Examples of oscillatory behavior appeared in the literature; see, e.g., [10], and the example in Figure 2 in [7].

There is a large body of literature on overload controls for Stored Program Control (SPC) systems. An extended list of references can be found, for example, in [20], where three basic overload controls of SPC-switches are discussed. It is shown that a certain priority discipline may cause severe oscillations in the system, creating large queues even if the system is underloaded. (See Section 4 in that reference.) Two important points are made, that are directly related to our work here: First, it is argued that the study of time-dependent (non-stationary) behavior of such stochastic systems during overload periods is an important research direction. Of course, fluid models should play a significant role in such settings, due to intractability of the stochastic transient behavior. Second, that any overload control should ensure that call delays do not increase during normal load periods. For example, a control must not generate oscillations in the manner described above, or create harmful congestion in the manner described in [25] and [34].

2 The Time-Varying X Model

We now describe the stochastic model in detail. The X depicted in Figure 1, has two customer classes and two agent pools, each with many agents. We assume that each customer class has a service pool primarily dedicated to it, but all agents are cross-trained so that they can handle calls from the other class, even though they may do so inefficiently, i.e., customers may be served at a slower rate when served in the other class

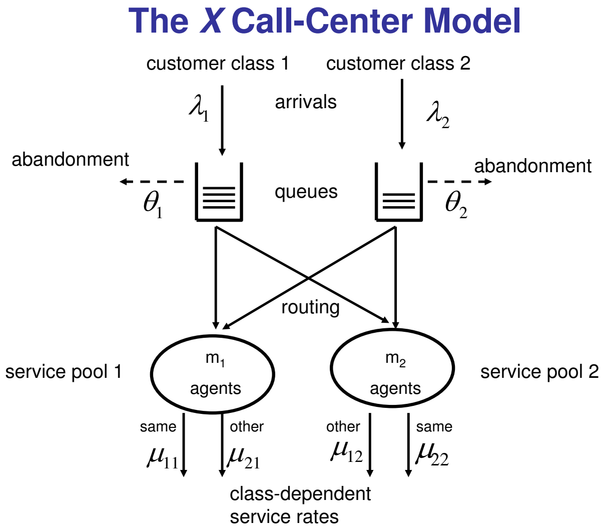


Figure 1: The X model

pool. We assume that both customer classes arrive according to independent *nonhomogeneous* Poisson processes with time-varying rate functions. The staffing levels are assumed to be time dependent as well, possibly anticipating the changes in the arrival rates. The arrival rates may be unknown, but we treat them as deterministic functions, i.e., we do not enforce any distributional assumptions on these rates. We let $\mu_{i,j}$ denote the service rate of class-i customers served in service pool j, $i, j = 1, 2$, and assume that service times are independent exponential random variables. Each class has an infinite buffer where customers who are not routed immediately into service upon their arrival are waiting for their turn to be served. Within each class, customers are served according to the first-come-first-served discipline. We further assume that each customer has a finite patience that is exponentially distributed, and will abandon if his waiting time in queue exceeds his (random) patience. We denote by θ_i the patience rate of class i, $i = 1, 2$.

Even though we do not prove any limit theorems here, and instead develop direct fluid models to approximate the stochastic system, we will use asymptotic considerations in our analysis. We therefore consider a sequence of X systems, as just described, indexed by a superscript n.

Specifically, for each $n \geq 1$ we let $\{\lambda_i^n(t) : t \geq 0\}$ denote the arrival-rate function to pool i, and $\{m_j^n(t) : t \geq 0\}$ denote the number of agents in pool j as a function of time, $i, j = 1, 2$. For the fluid approximation we assume that

$$\lambda_i^n(\cdot)/n \to \lambda_i(\cdot) \quad \text{and} \quad m_j^n(\cdot)/n \to m_j(\cdot) \quad \text{as } n \to \infty, \tag{1}$$

where $\lambda_i, m_j : [0, \infty) \mapsto [0, \infty)$, $i, j = 1, 2$. A rigorous treatment (which will not be carried out here) requires assumptions on the limit functions in (1). A minimal assumption, that is clearly not restrictive in applications, is that those limits are differentiable except possibly in a finite number of points. Therefore, the arrival rates and staffing functions can have points of discontinuity (have jumps), but the number of such points is finite. See Remark 5.1 below.

For all $n \geq 1$ let $Q_i^n(t)$ denote the number of customers waiting in the class-i buffer and $Z_{i,j}^n(t)$ be the number of class-i customers in service pool j at time t in system n. Define

$$X^n \equiv X^n(t) \equiv (Q_i^n(t), Z_{i,j}^n(t) : i, j = 1, 2), \quad t \geq 0. \tag{2}$$

If a control is employed that uses only information on the current state of X^n at each decision epoch, then X^n is a nonhomogeneous *continuous-time Markov chain* (CTMC).

Since we are interested in fluid analysis, it is reasonable to consider three possible regimes: underloaded, normally loaded and overloaded. We let $\{\rho_i^n(t) : t \geq 0\}$ denote the traffic-intensity function to pool i in system n, i.e., $\rho_i^n(t) := \lambda_i^n(t)/(\mu_{i,i} m_i^n(t))$, and consider the following limits $i = 1$ and $i = 2$.

$$\lim_{n\to\infty} (\rho_i^n(t) - 1) = \beta_i(t), \quad t \geq 0. \tag{3}$$

We say that pool i is underloaded at time t if $\beta_i(t) < 0$, overloaded at time t if $\beta(t) > 0$ and normally loaded at time t if $\beta_i(t) = 0$.

The interesting, and realistic, scenarios to investigate are when the system switches between different loads, e.g., from having both pools normally loaded to an overload case and back to normal, but remains at each regime for some non-negligible amount of time. We therefore assume that the system does not switch regimes too quickly, e.g., if pool 1 is overloaded at time t ($\beta_1(t) > 0$) and class-1 customers receive help from pool 2, then this overload incident and sharing take place for a few mean service times.

As we already mentioned above, it is typical in the call-center literature and in applications, to consider piecewise-constant arrival rate functions, and corresponding piecewise-constant staffing functions. We will thus consider three types of models:

(a) Stationary models. The arrival processes are homogeneous Poisson with fixed rates and constant staffing functions. We will assume that those functions have changed at time 0, and then consider the transient behavior of the system as it approaches a new steady state.

(b) Piecewise-stationary models. The arrival processes are nonhomogeneous Poisson processes with piecewise constant intensity (rate) functions. The staffing functions will also be assumed to be piecewise constants, and the system will switch between different loads.

(c) Time-varying model. The arrival rates and staffing functions are general functions of times, but, as mentioned above, we assume that they are continuously differentiable, except possibly on a finite set of points.

Clearly, the time-varying model (c) is the more general and includes the first two types of models. In models (b) and (c) the staffing functions may change in accordance to anticipated changes in the arrival rates. However, in all three models, the realized arrival rates may be different than anticipated, which can cause unexpected overloads.

2.1 Two Controls: FQR-T and FQR-ART

We now elaborate on the two controls which were already described briefly in §1. For each $n \geq 1$, the FQR-T control is based on two positive (activation) thresholds, $k_{1,2}^n$ and $k_{2,1}^n$ and the two queue-ratio parameters,

$r_{1,2}$ and $r_{2,1}$. We define two (centered) queue-difference stochastic processes

$$D^n_{1,2}(t) \equiv Q^n_1(t) - k^n_{1,2} - r_{1,2}Q^n_2(t) \quad \text{and} \quad D^n_{2,1}(t) \equiv r_{2,1}Q^n_2(t) - k^n_{2,1} - Q^n_1(t), \quad t \geq 0. \tag{4}$$

As long as $D^n_{1,2}(t) < 0$ and $D^n_{2,1}(t) < 0$ we consider the system to be not overloaded so that no customers are routed to be served in the other class' pool. Once one of these inequalities is violated, the system is considered to be overloaded, and sharing is initiated. For example, if $D^n_{1,2}(t) \geq 0$, then class 1 is judged to be overloaded (because then $Q^n_1 - r_{1,2}Q^n_2 \geq k^n_{1,2}$), and it is desirable to send class-1 customers to be served in pool 2. Note that $D^n_{1,2}(t) \geq 0$ does not exclude the case that class 2 is also overloaded; we can have $\beta_1(t), \beta_2(t) > 0$. However, once one of the thresholds is crossed, its corresponding class is considered to be "more overloaded" than the other class. We refer to this situation as *unbalanced overloads*. We refer to the thresholds $k^n_{1,2}$ and $k^n_{2,1}$ as *activation thresholds,* since the crossing of one of these thresholds activates sharing (although they also serve as a mean to prevent sharing when it is not required).

The behavior of X^n in (2) depends on the choice of the thresholds $k^n_{i,j}$. In particular, we want the thresholds to be large enough so that sharing will not take place if both service pools are normally loaded, and to be small enough to detect any overload quickly, and start sharing in the correct direction once the overload begins. Note that without sharing, the two pools operate as two independent $M/M/m^n + M$ (Erlang-A) models. The fluid and diffusion limits for the Erlang-A model give insight as to how to choose these thresholds; see [13]. In our previous papers we assumed that the activation thresholds are chosen to satisfy the following asymptotics:

$$k^n_{i,j}/n \to 0 \quad \text{and} \quad k^n_{i,j}/\sqrt{n} \to \infty \quad \text{as } n \to \infty, \quad i,j = 1,2. \tag{5}$$

The first limit in (5) ensures that overloads are detected quickly (immediately as $n \to \infty$) since the fluid limit of the queue in a pool that starts normally loaded (with no fluid queue) and becomes overloaded will become strictly positive immediately after the overload begins. The second limit in (5) ensures that stochastic fluctuations of normally-loaded pools will not cause undesired sharing, since the diffusion-scaled queue in that case are of order $\sqrt{n}$.

In the call-center settings, if sharing of customers takes place during overloads only, it is reasonable to assume that agents serve the other class customers (the so-called "shared customers") at a slower rate than their own designated customers. Thus, substantial sharing can reduce the effective service rate of the helping pool. To avoid sharing in both directions simultaneously, we enforced the one-way sharing rule described in §1. In the time-varying settings considered here, it is clear that the one-way sharing rule may considerably slow the response to a change in the direction of overloads. We elaborate in §3 below.

To remedy this problem, one can remove the one-way sharing rule altogether and rely solely on the activation thresholds to avoid undesired sharing. However, removing that rule can have harmful effects on the system. First, there is a need to choose activation thresholds that are higher than if the one-way sharing rule is employed, increasing the time until overloads are detected. Moreover, if these thresholds are too large, then some overloads may not be detected at all (recall that abandonment keep the queues from increasing indefinitely.) Second, if sharing is taking place in one direction, and then immediately starts in the other direction in response to a switch in the overload, then the combined service capacity of both pools may be reduced significantly, creating a period of severe congestion in both directions. Hence, it is beneficial to avoid too much simultaneous two-way sharing.

Adjusting the Control: FQR-ART. For the reasons discussed above, we suggest a modification of the one-way sharing rule by introducing *release thresholds* (RT). For each $n \geq 1$, we take two strictly positive numbers $\tau^n_{1,2}$ and $\tau^n_{2,1}$ and modify the control in the following manner: A newly available type-2 agent is allowed to take a class-1 customer at time t only if the number of type-1 agents serving class-2 customers at the same time t is below $\tau^n_{2,1}$ (and of course $D^n_{i,1}(t) \geq 0$), i.e., if $Z^n_{2,1}(t) \leq \tau^n_{2,1}$, and similarly in the other direction.

The new release thresholds make activation thresholds satisfying (5) not suitable for the time-varying environment, as will be shown in §4 below. Instead, these activation thresholds must be positive in "fluid scale", i.e., they should be chosen so as to satisfy

$$\lim_{n\to\infty} k^n_{i,j}/n = k_{i,j} > 0, \quad i,j = 1,2. \tag{6}$$

To summarize, the FQR-ART control is a modification of FQR-T in order to adjust for the time-varying environment considered here. For each system n, the FQR-ART control is specified by the six parameters $(r_{1,2}, r_{2,1}, k^n_{1,2}, k^n_{2,1}, \tau^n_{1,2}, \tau^n_{2,1})$ and the routing and scheduling rules which depend on the values of the two processes $D^n_{i,j}$ and $Z^n_{i,j}$, $i \neq j$, in the manner described above. We mention that FQR-T requires knowing only the values of the queues at each time t (specifically, the values of the two difference processes (4)), whereas FQR-ART also requires knowledge of $Z^n_{1,2}$ and $Z^n_{2,1}$. However, under either control, the X model is a (possible inhomogeneous) CTMC.

2.2 Analysis Via Fluid Approximations

Since the stochastic process X^n in (2) under FQR-ART is evidently too difficult to analyze exactly, we will employ a deterministic dynamical-system approximation, and refer to that approximation as "fluid approximation" or "fluid model" interchangeably. The main idea in using fluid approximations is that, for large n, $\bar{X}^n \approx x$, for some deterministic function x that is easier to analyze than the untractable stochastic process X^n. (We use the 'bar' notation throughout to denote fluid scaled processes, e.g., $\bar{X}^n \equiv X^n/n$.) In particular, the fluid counterpart of X^n in (2) is the six-dimensional function

$$x \equiv x(t) \equiv (q_i(t), z_{i,j}(t) : i,j = 1,2), \quad t \geq 0, \tag{7}$$

where q_i and $z_{i,j}$ are the fluid approximations for the stochastic processes Q^n_i and $Z^n_{i,j}$, $i,j = 1,2$.

The function x should formally arise as a *functional strong-law of large numbers* (FSLLN), and thus capture the time-dependent mean behavior of the process X^n.

Note that, in the stochastic system, routing of customers among the two pools depends on the values of the difference processes in (4). For example, if sharing is taking place with pool 2 helping class 1, and assuming $Z^n_{2,1} \leq \tau^n_{2,1}$, the process $D^n_{1,2}$ determines which customer class a newly available type-2 agent will take. Unfortunately, in the fluid system we cannot simply replace the process $D^n_{1,2}$ with a process

$$d_{1,2}(t) \equiv q_1(t) - k_{1,2} - r_{1,2}q_2(t), \quad t \geq 0.$$

In fact, the purpose of the control is to keep $d_{1,2}(t) = 0$ during the overload. Hence, to determine the evolution of the fluid model a refined analysis of the behavior of $D^n_{1,2}$ is required (or $D^n_{2,1}$ during overloads in the other direction). We elaborate in §5 below, where the fluid equations are developed.

3 Relaxing the One-Way Sharing Rule by Adding Release Thresholds

Relying on fluid considerations, we now demonstrate why the one-way sharing rule is not adequate to the current time-varying settings. As mentioned in §2.1 above, the system can be slow to respond to switching overloads if the one-way sharing is employed. The simple fluid analysis suggests that having release thresholds eliminates fixes that problem.

3.1 Drawbacks of One-Way Sharing in Time-Varying Settings

We consider two consecutive time intervals $I_0 = [t_0, t_1)$ and $I_1 = [t_1, t_2)$ with $0 \leq t_0 < t_1 < t_2 \leq \infty$, with the system being overloaded in opposite direction over each interval. Suppose, for example that over the time interval I_0 class 2 was overloaded and sharing took place with pool 1 helping class 2. Then, at time t_1 the loads have changed with class 1 becoming overloaded in such a way that sharing is required in the other direction. In particular, we assume that $\beta_1(t) > 0$ for $t \in I_1$ and that $z_{2,1}(t_1) > 0$.

Recalling that the fluid approximation has asymptotic significance, namely, that $z_{2,1}(0) > 0$ implies that $Z^n_{2,1}(0)$ is proportional to n (and n is large), the mean time to wait until pool 1 has no more class-2 customers is, for large n,

$$\sum_{j=1}^{Z^n_{2,1}(0)} \frac{1}{j \cdot \mu_{2,1}} \approx \frac{\log\left(Z^n_{2,1}(0)\right)}{\mu_{2,1}}, \tag{8}$$

where $\log(\cdot)$ is the natural logarithm, i.e., $\log(\cdot) \equiv \log_e(\cdot)$. We thus see that the time it takes a pool to empty from its shared customers is of order $\log(n)$ if n is large, when no new shared customers are routed to that pool.

A fluid approximation for the evolution of $Z^n_{1,2}$ can easily be derived using rate considerations. Since every type-1 agent who is helping a class-2 customer at time $t > t_1$ will finish service immediately after time t at a rate $\mu_{2,1}$, regardless of the value of t, due to the memoryless property, and since there are no more class 2 customers routed to pool 1 after time t_1, we expect that $z_{2,1}$ will satisfy the ODE

$$\dot{z}_{2,1}(t) = -\mu_{2,1} z_{2,1}(t), \quad t \in I_1,$$

whose unique solution is

$$z_{2,1}(t) = z_{2,1}(t_1)e^{-\mu_{2,1}t}, \quad t \in [t_1, t_2). \tag{9}$$

In particular, in the fluid model, if $z_{2,1}(t_1) > 0$, then pool 1 will never empty, so that sharing can never begin in the opposite direction.

3.2 Adding the Release Thresholds

The simple considerations leading to (8) and (9) show that a large system will be slow to react to changes in the direction of overloads, and will require an order of $\log(n)$ time units to switch the direction of sharing if the one-way sharing rule is employed.

However, the fluid model (9) suggests that, if $z_{2,1}(t_1) > \tau_{2,1}$ and $\tau_{2,1} > 0$, then $z_{2,1}$ will reach $\tau_{2,1}$ in finite time. In particular, $\tau_{2,1}$ will be hit at time

$$T \equiv \frac{1}{\mu_{2,1}} \log\left(\frac{z_{2,1}(t_1)}{\tau_{2,1}}\right).$$

This observation leads to the need of having release thresholds which are chosen as follows: For two strictly positive numbers $\tau_{1,2}$ and $\tau_{2,1}$ we take two sequences $\{\tau^n_{1,2}\}$ and $\{\tau^n_{2,1}\}$, satisfying

$$\tau^n_{1,2}/n \to \tau_{1,2} \quad \text{and} \quad \tau^n_{2,1}/n \to \tau_{2,1} \quad \text{as } n \to \infty. \tag{10}$$

Then, an available type-2 agent is allowed to serve a class-1 customer only if the proportion of type-1 agents serving class-2 customers is below $\tau^n_{2,1}$ (and of course $D^n_{i,1}(t) \geq 0$), and similarly in the other direction.

3.3 Simulation Experiments

To illustrate the importance of the release thresholds for (finite) stochastic systems, we compared the performance of a system with and without release thresholds in simulation. The results can be seen in Figures 2 and 3. The parameters for this simulation are

$$\begin{aligned} m^n_1 &= m^n_2 = 1000, \quad \lambda^n_1 = 1200, \quad \lambda^n_2 = 990, \quad \mu_{1,1} = \mu_{2,2} = 1, \quad \mu_{1,2} = \mu_{2,1} = 0.5, \\ \kappa^n_{1,2} &= \kappa^n_{2,1} = 100, \quad \text{and} \quad r = 1. \end{aligned}$$

(Here, we can think of n as being fixed and equal to 1000.) With these parameters, queue 1 is overloaded, while queue 2 is underloaded. To respond to that unbalanced overload by having pool 2 help class 1, we should have $Z^n_{1,2} > 0$ and $Z^n_{2,1} = 0$ if one-way sharing is employed. However, we initialize the system with sharing in the opposite way. We are interested in the time it takes the stochastic process $Z^n_{2,1}$ to reach 0, so that the desired sharing can begin. With release thresholds of only $\tau^n_{1,2} = \tau^n_{2,1} = 0.01n = 10$, that time is reduced from about 21 mean service times to about 9 service times. Thus, clearing the last 1% of the class-2 customers in pool 1 without release thresholds takes more than half the total clearing time.

Indeed, we consider an extreme example in which *all* of service pool 1 is initially busy with customers from class 2, and none of the type-2 agents are busy serving class-1 customers. We do this in order to convey the message that *it is the last few agents working with class* 1 *that cause the delayed response*. In particular, the $Z^n_{2,1}$ process decreases fast at the beginning, but then the decrease rate slows down considerably.

From Figures 2 and 3, it is also easy to see what happens in less extreme cases, when $0 < Z_{2,1}(0) <$

m_1. For example, if we initialize with 20% sharing in the wrong direction, we see that, without a release threshold, the time to activate sharing in the right direction is about $21 - 4 = 17$ time units. In contrast, with release thresholds, it is about $9 - 4 = 5$ time units. When we start with a lower percentage of agents sharing the wrong way, the difference becomes even more dramatic, because we eliminate a common initial period (here of length 4 time units).

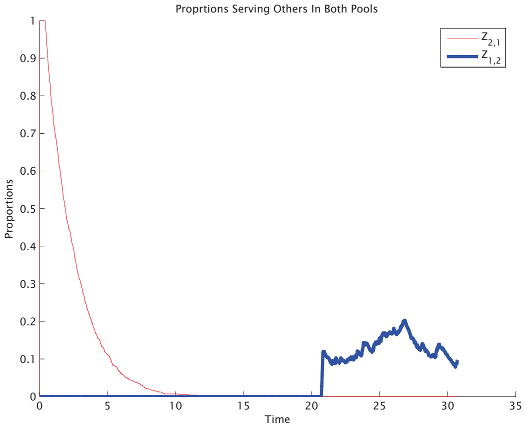


Figure 2: Sample paths of $Z_{1,2}(t)$ and $Z_{2,1}(t)$ initialized incorrectly, without release thresholds.

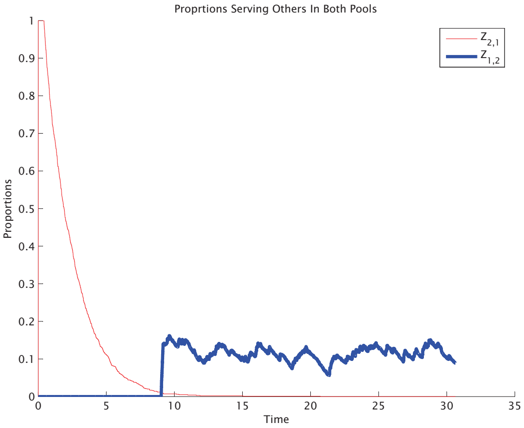


Figure 3: Sample paths of $Z_{1,2}(t)$ and $Z_{2,1}(t)$ initialized incorrectly, with release thresholds $\tau_{1,2} = \tau_{2,1} = 0.01$.

4 Congestion Collapse Due to Oscillations

The simple Fluid consideration in §3.1, and its application to the stochastic system, demonstrates the need for having release thresholds when the direction of overload switches. However, an overload period can also end with a return to normal loads, so that no sharing (in either direction) is required. We now show that the release thresholds can cause serious problems when the system returns to normal loading after an overload incident if the activation thresholds are chosen according to (5), as in the FQR-T control. The main problem that can arise in these setting is when the inefficient sharing condition holds, namely when

$$\mu_{1,1} > \mu_{2,1} \quad \text{and} \quad \mu_{2,2} > \mu_{1,2} \tag{11}$$

which is what we now assume.

Let time 0 be the time of that an overload period ends. For simplicity, we assume that both the arrival rates and staffing levels remain constat for all $t \geq 0$. For our purpose here, we further assume that

$$\lambda_1 = \mu_{1,1} m_1 \quad \text{and} \quad \lambda_2 = \mu_{2,2} m_2, \tag{12}$$

so that both pools would be normally loaded if there was no sharing. However, since we consider a system that is recovering from an overload, we assume that $z_{2,1}(0) > \tau_{2,1}$ (implying that $z_{2,1}(t) > 0$ for all $t > 0$

as well, by (9)). Now, for each time $t \geq 0$, the total effective service rate of pool 1 is

$$\mu_{1,1}(m_1 - z_{2,1}(t)) - \mu_{2,1} z_{2,1}(t) = \lambda_1 - (\mu_{1,1} - \mu_{2,1}) z_{2,1}(t) < \lambda_1,$$

where the equality follows from (12) and the strict inequality follows from (11) and the fact that $z_{2,1}(t) > 0$ for all $t \geq 0$. That implies that pool 1 is overloaded for all $t \geq 0$, even though it would have been normally loaded if there were no class-2 customers in the pool.

Let $t_0 \equiv \inf\{t \geq 0 : z_{2,1}(t) = \tau_{2,1}\}$. Since pool 1 is overloaded and no class-1 customers are sent to pool 2 on $[0, t_0)$, it holds that $q_1(t) > 0$ for all $t \in (0, t_0]$. If the number of class-1 customers in pool 2 is negligible initially, i.e., $z_{1,2}(0) = 0$, then q_2 will approach 0. Let $t_1 \equiv \inf\{t \geq 0 : q_1(t) - r_{1,2} q_2(t) = k_{1,2}\}$. If the activation thresholds are chosen according to (5), so that $k_{1,2} = 0$, then class-1 customers will be sent to pool 2 at time $t_2 \equiv \max\{t_0, t_1\}$, i.e., $z_{1,2}(t) > 0$ for all $t \geq t_2$. By the same reasoning given above, pool 2 is then overloaded for all $t \geq t_2$, so that *both service pools are overloaded for all* $t > t_2$.

This problem may seem marginal for finite systems if we choose $\tau_{i,j}$ to be sufficiently small (which we do). However, $\tau_{i,j}$ small only ensures that one of the pools is not “too overloaded” at any time, e.g., if at a time $t > t_2$, $z_{1,2}(t) \leq \tau_{1,2}$, then we still might have $z_{2,1}(t)$ relatively large, so that the effective service capacity of pool 1 is substantially reduced for a long time interval. The simple analysis above suggests that the system may start oscillating with $z_{i,j}$ growing large and then decreasing over and over again. Such oscillatory behavior may cause congestion collapse due to the effect it has on the long-run average service rate, namely, for some nonnegligible $C_{i,j} > \tau_{i,j}$, it may hold that

$$\liminf_{t \to \infty} \frac{1}{t} \int_0^t z_{i,j}(s) ds = C_{i,j} \quad i, j = 1, 2.$$

In particular, the long-run average service rate of the two pools combined can potentially be much lower than $\lambda_1 + \lambda_2$, making the system severely congested.

4.1 Simulations of Oscillating Systems

The oscillatory behavior of the system is hard to observe when there is abandonment. Thus, for display purposes, we start with an extreme case of a system with no abandonment, and then simulate a more realistic system with abandonment. Figures 4 and 5 show a single simulated sample path of a system case with $\mu_{1,2} = \mu_{2,1} = 0.1$ and no abandonment. The service rates for the designated classes are $\mu_{1,1} = \mu_{2,2} = 1$, there are 100 agents in each pool, and the arrival rates are $\lambda_1 = \lambda_2 = 98$, so that both pools are stable if there is no sharing. The two types of thresholds are taken to be $k^n_{i,j} = 10$ and $\tau^n_{i,j} = 1$, $i, j = 1, 2$. The symmetry of the system implies that both pools and queues exhibit the same behavior.

A more realistic example is shown in Figures 6 and 7, where $\mu_{1,2} = \mu_{2,1} = 0.5$ and the abandonment rates are $\theta_1 = \theta_2 = 0.5$. The other parameters are the same as in the previous extreme case. The oscillations of the service process are harder to observe, so we compare to a simulation of the exact same system, but with the activation thresholds increased to $k^n_{i,j} = 35$, shown in Figures 6 and 7.

Since the activation thresholds in FQR-ART should be of order n asymptotically, as in (6), we can use simple fluid considerations to compute good values for these thresholds in the fluid approximation, i.e., to

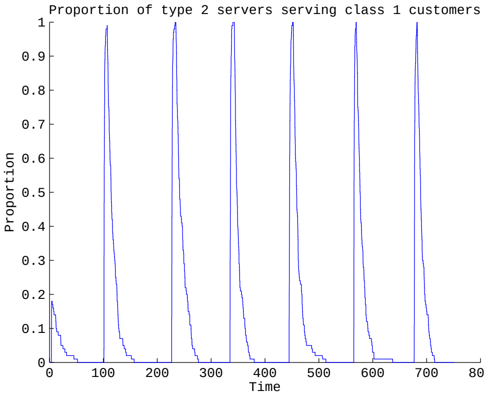


Figure 4: Oscillations of $Z_{1,2}$, extreme example: $\tau_{i,j} = 1$, $k_{i,j} = 10$, no abandonment.

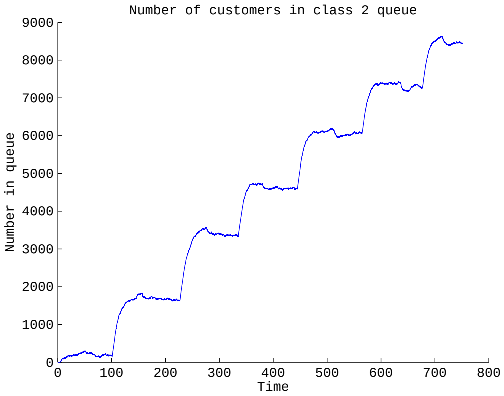


Figure 5: Oscillations of Q_2, extreme example: $\tau_{i,j} = k_{i,j} = 1$, no abandonment.

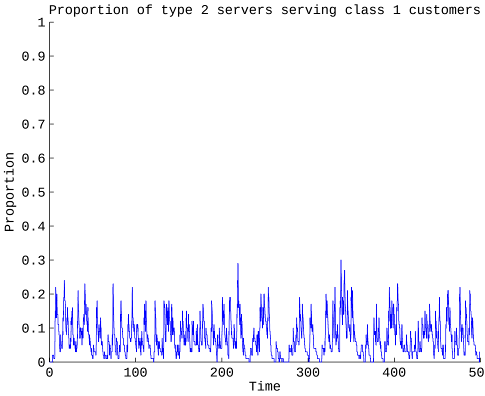


Figure 6: Oscillations of $Z_{1,2}$ with abandonments: $\tau_{i,j} = k_{i,j} = 1$.

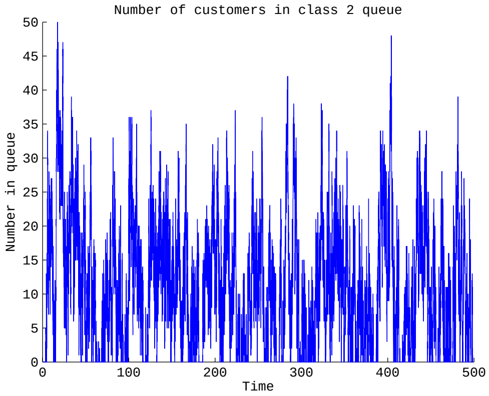


Figure 7: Oscillations of Q_2 with abandonments: $\tau_{i,j} = k_{i,j} = 1$.

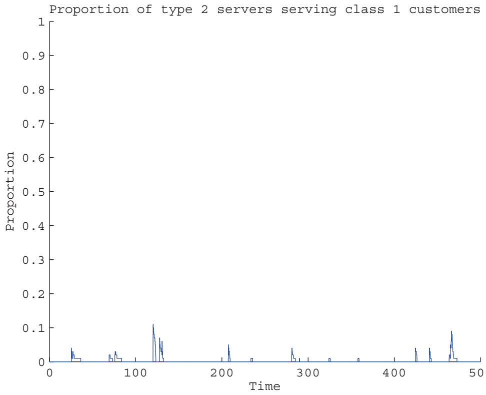


Figure 8: $Z_{1,2}$ with larger thresholds: $k_{i,j} = 35$.

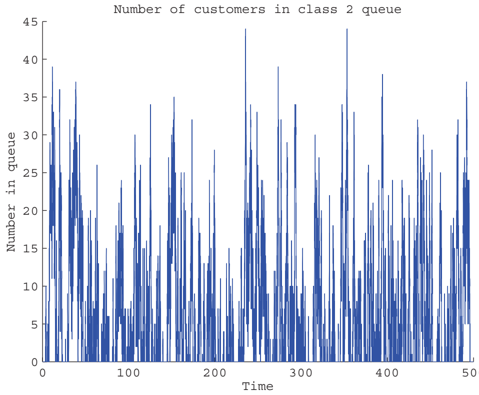


Figure 9: Q_2 with larger thresholds: $k_{i,j} = 35$.

find candidates for $k_{1,2}$ and $k_{2,1}$. (Engineering considerations are needed.) An example on how to choose the thresholds is given in the appendix.

5 The Fluid Model

The fluid model for the stochastic system X^n under FQR-ART is described implicitly as the solution to an ODE. In this section we derive that ODE via a heuristic representation of the inhomogeneous CTMC in (2).

5.1 Representation of the Stochastic System During Overloads

The sample paths of a queueing system are represented in terms of its primitive processes, i.e., the arrival, abandonment and service processes, as a function of the control. Unlike traditional fluid models, in which the primitive stochastic processes are replaced by their long-run rates, the deterministic fluid model here is more involved and includes a stochastic ingredient in the form of a stochastic AP, which we describe in detail in §5.2 below.

Even though we are not proving that the fluid model arises as a weak limit of the fluid-scaled stochastic system, we need to take asymptotic considerations in order to develop the fluid approximation. We thus start with a representation of the stochastic system during overloads, assuming that both service pools are full over an interval $[0, T]$, namely that

$$Z^n_{1,1}(t) + Z^n_{2,1}(t) = m^n_1(t) \quad \text{and} \quad Z^n_{2,2}(t) + Z^n_{1,2}(t) = m^n_2(t), \quad t \in [0, T]. \tag{13}$$

We next use random time-changes of independent unit-rate Poisson processes to represent the sample paths of X^n, as reviewed in [24]; see Equations (41)-(43) in [28] for such a representation applied to the X model operating under FQR-T. For example, the representation of Q^n_1 over $[0, T]$ is

$$\begin{aligned}
Q^n_1(t) = {} & N^a_1\left(\int_0^t \lambda^n_1(s)ds\right) - N^u_1\left(\theta_1 \int_0^t Q^n_1(s)ds\right) \\
& - N^+_1\left(\int_0^t \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\le\tau^n_{2,1}\}\}} \left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{1,2}Z^n_{1,2}(s) + \mu_{2,1}Z^n_{2,1}(s) + \mu_{2,2}Z^n_{2,2}(s)\right) ds\right) \\
& - N^-_1\left(\int_0^t (1 - \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\le\tau^n_{2,1}\}\}} - \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\le\tau^n_{1,2}\}\}}) \left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{2,1}Z^n_{2,1}(s)\right) ds\right),
\end{aligned}$$

where N^a_1, N^u_1, N^+_1 and N^-_1 are mutually independent unit rate (homogeneous) Poisson processes, and $\mathbf{1}_A$ is the indicator function that is equal to 1 if even A occurs.

Note that the representation of Q^n_1 is essentially a flow equation (based on the memoryless property of the exponential distribution). That is, the queue at time t is all those customers who arrived by that time, captured by the Poisson process N^a_1, minus all the customers that abandoned, captured by the Poisson process N^u_1, minus all those who were routed into service, as captured by the last two Poisson processes in the expression.

We elaborate on how the intensities of the last two Poisson processes in the right-hand side (RHS) of the representation were obtained. First, if at time $s \in [0, T]$ the event $\mathcal{D}_{1,2}(s) \equiv \{D^n_{1,2}(s) > 0 \cap Z^n_{2,1}(s) \le \tau^n_{2,1}\}$ holds, then any newly available agent in the system will take his next customer from the head of queue 1. Since agents become available at an instantaneous rate $\sum_{i,j} \mu_{i,j} Z^n_{i,j}(s)$ at time s, we get the third component in the RHS of $Q^n_1(t)$. Next we recall that, by the routing rule of FQR-ART, if at a time $s \in [0, T]$

$\mathcal{D}_{2,1}(s) \equiv \{(D^n_{2,1}(s) > 0) \cap (Z^n_{1,2}(s) \leq \tau^n_{1,2})\}$ holds, then any newly available agent takes his next customer from queue 2, in which case queue 1 will not decrease due to a service completion. If neither of the events $\mathcal{D}_{1,2}(s)$ or $\mathcal{D}_{2,1}(s)$ holds at a time s, then only service completions at pool 1 will cause a decrease at queue 1 due to a customer from that queue being routed to service. That explains the last term in the RHS of the representation.

Next, we exploit the fact that each of the Poisson processes in the representation minus its random intensity constitutes a martingale (again, see [24]), e.g.,

$$M^{n,u}_1 \equiv N^u_1\left(\theta_1 \int_0^t Q^n_1(s)ds\right) - \theta_1 \int_0^t Q^n_1(s)ds$$

is a martingale. Thus, subtracting and then adding all the random intensities, and using the fact that a sum of martingales is again a martingale, we get the following representation for the processes $Q^n_1, Q^n_2, Z^n_{1,2}, Z^n_{2,1}$ (the remaining two processes $Z^n_{1,1}$ and $Z^n_{2,2}$ are determined by (13)).

$$\begin{aligned}
Q^n_1(t) &= M^n_1(t) + \int_0^t \lambda^n_1(s)ds - \int_0^t \theta_1 Q^n_1(s)ds \\
&\quad - \int_0^t \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}} \left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{1,2}Z^n_{1,2}(s) + \mu_{2,1}Z^n_{2,1}(s) + \mu_{2,2}Z^n_{2,2}(s)\right) ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}} - \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\leq\tau^n_{1,2}\}\}}) \left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{2,1}Z^n_{2,1}(s)\right) ds, \\
Q^n_2(t) &= M^n_2(t) + \int_0^t \lambda^n_2(s)ds - \int_0^t \theta_2 Q^n_2(s)ds \\
&\quad - \int_0^t \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\leq\tau^n_{1,2}\}\}} \left(\mu_{1,1}Z^n_{1,1}(s) + \mu_{1,2}Z^n_{1,2}(s) + \mu_{2,1}Z^n_{2,1}(s) + \mu_{2,2}Z^n_{2,2}(s)\right) ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}} - \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\leq\tau^n_{1,2}\}\}}) \left(\mu_{2,2}Z^n_{2,2}(s) + \mu_{1,2}Z^n_{1,2}(s)\right) ds, \\
Z^n_{1,2}(t) &= M^n_{1,2}(t) + \int_0^t \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}} \mu_{2,2}Z^n_{2,2}(s)ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}}) \mu_{1,2}Z^n_{1,2}(s)ds, \\
Z^n_{2,1}(t) &= M^n_{2,1}(t) + \int_0^t 1_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\leq\tau^n_{2,1}\}\}} \mu_{1,1}Z^n_{1,1}(s)ds \\
&\quad - \int_0^t (1 - \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\leq\tau^n_{1,2}\}\}}) Z^n_{2,1}(s))ds,
\end{aligned} \tag{14}$$

where $M^n_1, M^n_2, M^n_{1,2}$ and $M^n_{2,1}$ are the martingale terms alluded to above. It is not hard to show that those martingales are negligible in the fluid scaling, i.e., that $M^n_i \Rightarrow 0$ and $M^n_{i,j} \Rightarrow 0$ as $n \to \infty$, uniformly over $[0, T]$, $i, j = 1, 2$; see, e.g., Lemma 6.1 in [28]. Hence, we consider those martingales as a negligible stochastic noise that can be ignored for the purpose of developing the fluid approximation for (14).

To replace the stochastic integral representation in (14) with a deterministic one, we need to replace the indicator functions with smooth functions. We start by assuming that there is a fluid counterpart x for X^n in (14) which is *continuous and differentiable*. (This fact can be shown to hold by a minor modification of Corollary 5.1 in [28]). For any fluid point $x(t)$, let

$$d_{1,2}(x(t)) \equiv q_1(t) - r_{1,2}q_2(t) - k_{1,2} \quad \text{and} \quad d_{2,1}(x(t)) \equiv r_{2,1}q_2(t) - q_1(t) - k_{2,1}. \tag{15}$$

We first observe that, if $d_{i,j}(x(t)) > 0$ then, since $d_{i,j}(\cdot)$ is a continuous function, $d_{i,j}$ is strictly positive over an interval, and similarly if $d_{i,j} < 0$, $i, j = 1, 2$. In such cases the indicator functions are easy to deal with because each is a constant over the interval, and equals either 1 or 0. For example, if $d_{1,2}(x(t)) > 0$ for $t \in [s_1, s_2)$, for some $0 \le s_1 < s_2 < \infty$, and in addition, $Z^n_{2,1}(t) \le \tau^n_{2,1}$ over that interval for all n large enough, then

$$\mathbf{1}_{\{\{D^n_{1,2}(t)>0\}\cap\{Z^n_{2,1}(t)\le\tau^n_{2,1}\}\}} = \mathbf{1}_{\{[s_2,s_2)\}} \quad \text{for all } n \text{ large enough}.$$

Hence, a careful study is required for all $x(t) \equiv \gamma$ in the *boundary sets* defined by

$$\mathbb{B}_{1,2} \equiv \{\gamma \in \mathbb{R}_6 : d_{1,2}(\gamma) = 0\} \quad \text{and} \quad \mathbb{B}_{2,1} \equiv \{\gamma \in \mathbb{R}_6 : d_{2,1}(\gamma) = 0\} \tag{16}$$

Note that FQR-ART aims to “pull” the fluid model to one of these two boundary sets during overloads, when sharing is actively taking place, i.e., $\mathbb{B}_{i,j}$ is the region of the state space where we aim the fluid model to be when pool j helps class i, $i, j = 1, 2$.

Unfortunately, there is no straightforward fluid counterpart to the stochastic processes $D^n_{1,2}$ and $D^n_{2,1}$ when the fluid is in the boundary sets. However, there are two related stochastic processes, operating in an infinitely faster time scale, whose behavior determines the evolution of the fluid model, as we now explain.

5.2 A Stochastic Averaging Principle

Before we explain how to deal with the indicator functions in the representation (14), we emphasize that the following explanation is for the purpose of gaining insight only. The explanation draws on results in [28], which were proved in different settings than here.

Assume, for example, that $x(t) \in \mathbb{B}_{1,2}$ and consider $D^n_{1,2}$. To be able to apply the results in [28], we assume (for now) that the arrival rates are fixed (the arrival processes are homogeneous Poisson processes) and that $Z^n_{2,1} < \tau_{2,1}$, so that routing is determined solely on the value of $D^n_{1,2}$. In particular, sharing can take place if $D^n_{1,2}(t) > 0$. Then, by Theorem 4.5 in [28],

$$D^n_{1,2}(t) \Rightarrow D_{1,2}(x(t), \infty) \quad \text{in } \mathbb{R} \quad \text{as } n \to \infty, \tag{17}$$

where $D_{1,2}(\gamma, \cdot) \equiv \{D_{1,2}(\gamma, s) : s \ge 0\}$ is a CTMC associated with $\gamma \in \mathbb{R}_6$ whose distribution is determined by the value γ. (There is a different process for each γ.)

An analogous result holds for $D^n_{2,1}$ when $x(t) \in \mathbb{B}_{2,1}$. The notation $D_{i,j}(\gamma, \infty)$ stands for a random variable that has the steady-state distribution of the CTMC $D_{i,j}(\gamma, \cdot)$. Loosely speaking, $D^n_{i,j}$ moves so fast when $x(t)$ is in $\mathbb{B}_{i,j}$, that it reaches its steady state instantaneously as $n \to \infty$. Hence, we call $D_{i,j}(\gamma, \cdot)$ the *fast-time-scale process* (FTSP) associated with the point γ, or simply the FTSP.

Since we are interested in analyzing the indicator functions in (20), we first define

$$D_{i,j}(\gamma, \cdot) \equiv +\infty \quad \text{if} \quad d_{i,j}(\gamma) > 0 \quad \text{and} \quad D_{i,j}(\gamma, \cdot) \equiv -\infty \quad \text{if} \quad d_{i,j}(\gamma) < 0, \quad \gamma \in \mathbb{R}_6.$$

Next, we define

$$\begin{aligned} \pi_{1,2}(\gamma) &\equiv P(D_{1,2}(\gamma, \infty) > 0), \quad \text{for} \quad \gamma \in \mathbb{B}_{1,2} \quad \text{and} \\ \pi_{2,1}(\gamma) &\equiv P(D_{2,1}(\gamma, \infty) > 0), \quad \text{for} \quad \gamma \in \mathbb{B}_{2,1}. \end{aligned} \tag{18}$$

Now, by Theorem 4.1 in [28], which was proved for the process $D^n_{1,2}$ when $x \in \mathbb{B}_{1,2}$, and assuming that $Z^n_{2,1}(s) \le \tau^n_{2,1}$ over $[t_1, t_2]$ for all n large enough, we have that

$$\int_{t_1}^{t_2} \mathbf{1}_{\{\{D^n_{1,2}(s)>0\}\cap\{Z^n_{2,1}(s)\le\tau^n_{2,1}\}\}} ds \Rightarrow \int_{t_1}^{t_2} \pi_{1,2}(x(s)) ds.$$

Similarly, if $x \in \mathbb{B}_{2,1}$ over an interval $[t_3, t_4]$, and $Z^n_{1,2}(s) \le \tau^n_{1,2}$ for all n large enough over that interval, we have

$$\int_{t_3}^{t_4} \mathbf{1}_{\{\{D^n_{2,1}(s)>0\}\cap\{Z^n_{1,2}(s)\le\tau^n_{1,2}\}\}} ds \Rightarrow \int_{t_3}^{t_4} \pi_{1,2}(x(s)) ds.$$

The convergence in both equations above holds uniformly.

We named this latter result a “stochastic averaging principle”, or simply an *averaging principle* (AP), since the process $D^n_{i,j}(t)$ is replaced by the *long-run average* behavior of the corresponding FTSP $D_{i,j}(x(t), \cdot)$ for each time t over the appropriate interval.

In the FQR-ART settings, the AP holds under the assumption that $Z^n_{i,j}$ lies below the appropriate release threshold over the interval $[t_1, t_2]$ for all n large enough (i.e., with probability converging to 1 as $n \to \infty$). If $Z^n_{i,j}$ is above the appropriate release threshold for all n large enough (again, with probability converging to 1) over $[t_1, t_2]$, then the limit of the integral considered above is clearly the 0 function. It remains to rigorously prove convergence theorems at points at which $Z^n_{i,j}(t) = \tau^n_{i,j} + o_P(n)$, where $o_P(n)$ denotes a random variable satisfying $o_P(n)/n \Rightarrow 0$ as $n \to \infty$. However, it is not hard to guess the dynamics of the limit (if it exists) at such points, as we do in our fluid approximation below.

5.3 Representation via an ODE

The above limiting arguments lead to the following fluid approximation for the X system under FQR-ART during overload periods. Considering an interval $[0, T]$ for which

$$z_{1,1}(t) + z_{2,1}(t) = m_1(t) \quad \text{and} \quad z_{2,2}(t) + z_{2,2}(t) = m_2(t) \quad \text{for all} \quad t \in [0, T], \tag{19}$$

together with an initial condition $x(0)$, the fluid model of X^n is the solution $x \equiv \{x(t) : t \ge 0\}$ over $[0, T]$ to the ODE:

$$
\begin{aligned}
\dot{q}_1(t) &= \lambda_1(t) - \theta_1 q_1(t) - \Pi_{1,2}(x(t))\left(\mu_{1,1}z_{1,1}(t) + \mu_{1,2}z_{1,2}(t) + \mu_{2,1}z_{2,1}(t) + \mu_{2,2}z_{2,2}(t)\right) \\
&\quad - \left(1 - \Pi_{1,2}(x(t)) - \Pi_{2,1}(x(t))\right)\left(\mu_{1,1}z_{1,1}(t) + \mu_{2,1}z_{2,1}(t)\right), \\
\dot{q}_2(t) &= \lambda_2(t) - \theta_2 q_2(t) - \Pi_{2,1}(x(t))\left(\mu_{1,1}z_{1,1}(t) + \mu_{1,2}z_{1,2}(t) + \mu_{2,1}z_{2,1}(t) + \mu_{2,2}z_{2,2}(t)\right) \\
&\quad - \left(1 - \Pi_{1,2}(x(t)) - \Pi_{2,1}(x(t))\right)\left(\mu_{2,2}z_{2,2}(t) + \mu_{1,2}z_{1,2}(t)\right), \\
\dot{z}_{1,2}(t) &= \Pi_{1,2}(x(t))\mu_{2,2}z_{2,2}(t) - (1 - \Pi_{1,2}(x(t)))\mu_{1,2}z_{1,2}(t), \\
\dot{z}_{2,1}(t) &= \Pi_{2,1}(x(t))\mu_{1,1}z_{1,1}(t) - (1 - \Pi_{2,1}(x(t)))\mu_{2,1}z_{2,1}(t), \\
\dot{m}_1(t) &= \dot{z}_{1,1}(t) + \dot{z}_{1,2}(t), \\
\dot{m}_2(t) &= \dot{z}_{2,2}(t) + \dot{z}_{2,1}(t).
\end{aligned}
\tag{20}
$$

where, for $\pi_{i,j}(x(t))$ in (18), $i, j = 1, 2$,

$$
\Pi_{i,j}(x(t)) := \begin{cases} \pi_{i,j}(x(t)) & \text{if } z_{j,i}(t) < \tau_{j,i}, \\ 0 & \text{otherwise.} \end{cases}
$$

Note that the ODE (20) can be equivalently represented by an integral equation resembling (14), but with the negligible martingale terms omitted, all the stochastic processes replaced by their fluid counterparts, and the indicator functions replaced by the appropriate $\Pi_{i,j}$ functions.

5.4 The Fluid Model When There is No Active Sharing

The ODE for the fluid model above was developed for all cases for which both pools are full, i.e., (19) holds. This is the main case because systems are typically designed to operate with very little extra service capacity (if any), and is clearly significant when overloads occur. In particular, note the a normally-loaded system, with $\beta_1(t) = \beta_2(t) = 0$, will have the two pools full, at least after some short time period. Since the system may go through periods in which at least one of the pools is underloaded, we now briefly describe the fluid models for underloaded pools.

Consider an interval $I \subset [0, \infty)$. If no sharing takes place and $z_{1,2}(t) = z_{2,1}(t) = 0$ for all $t \in I$, then the two classes operate as two independent Erlang-A models over that interval I, to which fluid limits are easy to establish. Specifically, assuming without loss of generality, that $I = [0, s)$ for some $0 < s < \infty$, the fluid dynamics of both classes obey the ODE

$$
\begin{aligned}
\dot{q}_i(t) &= (\lambda_i(t) - \mu_{i,i}z_{i,i}(t) - \theta_i q_i(t))\mathbf{1}_{\{q_i(t) \geq 0\}} \\
\dot{z}_{i,i} &= \begin{cases} 0 & \text{if } q_i(t) > 0, \\ \lambda_i(t)\mathbf{1}_{\{z_{i,i}(t) \leq m_i(t)\}} - \mu_{i,i}z_{i,i}(t) & \text{if } q_i(t) = 0. \end{cases}
\end{aligned}
\tag{21}
$$

In the time-invariant case, the unique solution for a given initial condition to the ODE in (21) is easily seen to be

$$q_i(t) = \left(\frac{\lambda_i - \mu_{i,i} m_i}{\theta_i} + \left(q_i(0) - \frac{\lambda_i - \mu_{i,i} m_i}{\theta_i} \right) e^{-\theta_i t} \right) \vee 0,$$
$$z_{i,i}(t) = \begin{cases} m_{i,i} & \text{if } q_i(t) > 0, \\ \frac{\lambda_i}{\mu_{i,i}} + \left(z_{i,i}(0) - \frac{\lambda_i}{\mu_{i,i}} \right) e^{-\mu_{i,i} t} & \text{if } q_i(t) = 0. \end{cases} \tag{22}$$

where $a \vee b \equiv \max\{a, b\}$ and $(q_1(0), q_2(0), z_{1,1}(0), z_{2,2}(0))$ is a deterministic vector in $[0, \infty)^2 \times [0, m_1] \times [0, m_2]$.

If $z_{1,2}(s_0) > 0$ (or $z_{2,1}(s_0) > 0$) for some $s_0 \geq 0$ and there is no active sharing over the interval $[s_0, s_1)$, then $z_{1,2}$ ($z_{2,1}$) is strictly decreasing over that interval. Then $z_{i,j}$, $i \neq j$, satisfies the ODE

$$\dot{z}_{i,j}(t) = -\mu_{i,j} z_{i,j}(t), \quad s_0 \leq t < s_1$$

which is the same as the ODE for $z_{i,j}$ in (20) with $\Pi_{i,j} = 0$.

Remark 5.1. A proof of existence of a unique solution to the ODE (20) requires showing that the RHS is a local Lipschitz continuous function of x and is piecewise continuous in t. We do not prove such a result here, but it is important to consider arrival rates and staffing functions that ensure that the RHS of the ODE satisfies the piecewise continuity condition in the time argument.

6 Solving the ODE

To compute the solution to (20) requires a computation of $\pi_{1,2}(x(t))$ and $\pi_{2,1}(x(t))$ for all $x(t) \in \mathbb{R}_6$. Simplification is achieved when $r_{1,2} = r_{2,1} = 1$, because the FTSP's $D_{i,j}(x(t), \cdot)$, $i, j = 1, 2$, become simple *birth-and-death* (BD) processes. To facilitate the discussion we thus consider this simpler case and refer to §6.2 in [27] for the treatment of the FTSP $D_{1,2}$ as a *quasi-birth-and-death process* (QBD) when the ratio parameters are not equal to 1. (In [27] FQR-T is studied with one overload incident, with pool 1 receiving help, but the same method can be applied to $D_{2,1}$ with sharing in the opposite direction.)

For simplicity, we again start by assuming that the arrival processes are homogenous Poisson processes, having constant arrival rates λ_1 and λ_2 over $[0, T]$, and that the staffing functions are also fixed over that time interval at m_1 and m_2. Recall that $D_{i,j}(\gamma, \cdot) \equiv \infty$ if $d_{i,j}(\gamma) > 0$ and $D_{i,j}(\gamma, \cdot) \equiv -\infty$ if $d_{i,j}(\gamma) < 0$, and let $\mathbb{A}_{1,2}$ and $\mathbb{A}_{2,1}$ be the subsets of $\mathbb{R}_6$ in which the FTSP's $D_{1,2}(\gamma, \cdot)$ and $D_{2,1}(\gamma, \cdot)$ are positive recurrent, i.e.,

$$\mathbb{A}_{1,2} \equiv \{\gamma \in \mathbb{B}_{1,2} : 0 < \pi_{1,2}(\gamma) < 1\} \quad \text{and} \quad \mathbb{A}_{2,1} \equiv \{\gamma \in \mathbb{B}_{2,1} : 0 < \pi_{2,1}(\gamma) < 1\}. \tag{23}$$

By definition, if the fluid model at time t is in $\mathbb{A}_{i,j}$, i.e., $x(t) \in \mathbb{A}_{i,j}$, then $d_{i,j}(x(t)) = 0$. However, if $d_{i,j}(x(t)) = 0$, then $x(t)$ is not necessarily in $\mathbb{A}_{i,j}$, because the FTSP $D_{i,j}(x(t), \cdot)$ may be transient (drift to $+\infty$ or $-\infty$) or null recurrent; in particular, *The evolution of the fluid model is determined by the distributional characteristics of the FTSP's $D_{1,2}$ and $D_{2,1}$*. Hence, even before we try to compute $\pi_{i,j}(x(t))$, which is necessary in order to solve the ODE (20), there is a need to determine whether $x(t)$ is in one of the sets $\mathbb{A}_{1,2}$ or $\mathbb{A}_{2,1}$. We focus on $D_{1,2}$, with the analysis of $D_{2,1}$ being similar.

To determined the behavior of the FTSP $D_{1,2}$ it is again helpful to think of x as a fluid limit of the fluid-scaled sequence $\{\bar{X}^n : n \geq 1\}$ and to recall that $D_{1,2}$ was achieved as a limit of $D^n_{1,2}$ without any scaling; see (17). (See also Theorem 4.4 in [28] which provides a process-level limit relating $D_{1,2}$ and $D^n_{1,2}$.) Hence, both processes are defined on the same state space, which, for $r_{1,2} = 1$, is $\mathbb{Z} \equiv \{\dots, -1, 0, 1, \dots\}$.

Now, for a fixed $x(t)$, when $D_{1,2}(x(t), \cdot) = m > 0$, the birth and death rates of the FTSP are, respectively,

$$\begin{aligned}
&\lambda^+(x(t), m) \equiv \lambda_1(t) + \theta_2 q_2(t), \\
&\mu^+(x(t), m) \equiv \lambda_2(t) + \mu_{1,1} z_{1,1}(t) + \mu_{1,2} z_{1,2}(t) + \mu_{2,1} z_{2,1}(t) + \mu_{2,2} z_{2,2}(t) + \theta_1 q_1(t).
\end{aligned}$$

In analogy to the (non-Markov) process $D^n_{1,2} = Q^n_1 - Q^n_2 - k^n_{1,2}$, $\lambda_+(x(t), m)$ corresponds to an increase of $D_{1,2}$ due to arrival to queue 1 plus an abandonment from queue 2 (since either one of these two events cause an increase by 1 of $D^n_{1,2}$ in the stochastic system). Since any other event causes $D^n_{1,2}$ to decrease by 1, due to the scheduling rules of FQR-ART, we get the expression for $\mu^+(x(t), m)$.

Next, if $D_{1,2}(x(t), m) = m \leq 0$, the birth and death rates are, respectively,

$$\begin{aligned}
\lambda^-(x(t), m) &\equiv \lambda_1 + \mu_{2,2} z_{2,2}(t) + \mu_{1,2} z_{1,2}(t) + \theta_2 q_2(t), \\
\mu^-(x(t), m) &\equiv \lambda_2 + \mu_{1,1} z_{1,1}(t) + \mu_{2,1} z_{2,1}(t) + \theta_1 q_1(t).
\end{aligned}$$

Again, whenever $D^n_{1,2}$ is non-positive and sharing is taking place with pool 2 helping class 1, a “birth” occurs if there is an arrival to queue 1 or an abandonment from queue 2, or if there is a service completion in pool 2 (since then a newly available type-2 agent takes his next customer from queue 2). Similarly, a “death” occurs if there is an arrival to class 2, an abandonment from queue 1, or a service completion in pool 1.

We see that the FTSP $D_{1,2}(x(t), \cdot)$ is a two-sided $M/M/1$ queue, i.e., it behaves like an $M/M/1$ queue with “arrival rate” $\lambda^+(x(t), m)$ and “service rate” $\mu^+(x(t), m)$ for all $m > 0$, and behaves like a different $M/M/1$ queue with “arrival rate” $\mu^-(x(t), m)$ and “service rate” $\lambda^-(x(t), m)$, for all $m \leq 0$. Thus, for

$$\delta^+(\gamma) \equiv \lambda^+(\gamma, \cdot) - \mu^+(\gamma, \cdot) \quad \text{and} \quad \delta^-(\gamma) \equiv \lambda^-(\gamma, \cdot) - \mu^-(\gamma, \cdot), \quad \gamma \in \mathbb{B}_{1,2},$$

the set $\mathbb{A}_{1,2}$ can be characterized via

$$\mathbb{A}_{1,2} \equiv \{\gamma \in \mathbb{B}_{1,2} : \delta^+(\gamma) < 0 < \delta^-(\gamma)\}.$$

Next, letting $T^+(\gamma)$ and $T^-(\gamma)$ denote, respectively, the busy period of the $M/M/1$ in the positive region and the busy period of the $M/M/1$ in the negative region, and using simple alternating renewal arguments for the renewal process $D_{1,2}(\gamma, \cdot)$, we have

$$\pi_{1,2}(\gamma) = \frac{E[T^+(\gamma)]}{E[T^+(\gamma)] + E[T^-(\gamma)]}, \tag{24}$$

where, from basic $M/M/1$ theory,

$$E[T^{\pm}(\gamma)] = \frac{1}{\mu^{\pm}(\gamma) - \lambda^{\pm}(\gamma)}.$$

Note that if $d_{1,2}(\gamma) = 0$ but $\gamma \notin \mathbb{A}_{1,2}$, then $\pi_{1,2}(\gamma)$ is equal to either 1 or 0. In particular,

$$\text{if } \delta^{+}(\gamma) \geq 0, \text{ then } \pi_{1,2}(\gamma) = 1 \text{ and if } \delta^{-}(\gamma) \leq 0 \text{ then } \pi_{1,2}(\gamma) = 0. \quad (25)$$

There are no other options, since for any $\gamma = x(t)$ for which both pools are full (as is required for the ODE (20) to be valid), it holds that

$$\delta^{-}(x(t)) - \delta^{+}(x(t)) = 2(\mu_{1,2} z_{1,2}(t) + \mu_{2,2} z_{2,2}(t)) > 0,$$

where the inequality above follows from the fact that $z_{1,2}(t) + z_{2,2}(t) = m_2(t) > 0$.

We see that the sets $\mathbb{A}_{i,j}$ and the computation of $\pi_{i,j}(\cdot)$ are completely determined by the staffing, arrival rates, service and abandonment rates for any given point $\gamma \in \mathbb{R}_6$, where the only points that require careful analysis are those in one of the two sets $\mathbb{B}_{i,j}$. However, if the arrival rates or the staffing functions are time dependent, then the distribution of the FTSP $D_{i,j}(x(t), \cdot)$ is also time dependent. In particular, given a $\gamma \in \mathbb{R}_6$ we cannot determine whether $D_{1,2}(\gamma, \cdot)$ is positive recurrent or not, since that may depend on the time $t \in [0, T]$. Thus, the sets at which the FTSP's are ergodic are themselves time dependent, and we need to consider sets of the form $\{\mathbb{A}_{i,j}(t) : t \in [0, T]\}$, where

$$\mathbb{A}_{i,j}(t) \equiv \{(\gamma, t) \in \mathbb{B}_{i,j} \times \mathbb{R}_{+} : \delta^{+}(\gamma, t) < 0 < \delta^{-}(\gamma, t)\}, \quad (26)$$

where $\delta^{+}(\gamma, t)$ and $\delta^{-}(\gamma, t)$ are the drifts of the FTSP $D_{1,2}(\gamma, \cdot)$ at the point γ at time t.

Fortunately, for the purpose of solving the ODE, we do not actually need to characterize the sets $\{\mathbb{A}_{i,j}(t) : t \in [0, T]\}$, because we can determine whether $D_{i,j}(x(t), \cdot)$ is ergodic at each time t as we solve the ODE.

6.1 A Numerical Algorithm to Solve the ODE

We can use the analysis in §6 to numerically solve the ODE (20), starting at a given initial condition $x(0)$, since we can now determine the value of $\Pi_{i,j}(x(t))$ for each $t \geq 0$. For example, if at a time $t \geq 0$ $d_{1,2}(x(t)) = 0$, then we check whether (26) holds, so that $x(t) \in \mathbb{A}_{1,2}(t)$. If $z_{2,1}(t) \leq \tau_{2,1}$, then $\Pi_{1,2}(x(t)) = \pi_{1,2}(x(t))$ and it can be computed using (24). If $z_{2,1}(t) > \tau_{2,1}$, then $\Pi_{2,1}(t) = 0$. If $d_{1,2}(x(t)) = 0$ but $x(t) \notin \mathbb{A}_{1,2}(t)$, i.e., if (26) does not hold, then we can determine the value of $\pi_{1,2}(x(t))$, and thus of $\Pi_{1,2}(x(t))$, by computing the drifts of the FTSP and employing (25) (replacing the drifts in (25) with the time dependent drifts as in (26)). Similarly we can compute the value of $\Pi_{2,1}(x(t))$ whenever $d_{2,1}(x(t)) = 0$.

In all other regions of the state space for which both pools are full, i.e., $z_{i,j}(t) + z_{j,i}(t) = m_j(t)$, $i \neq j$, we can easily determine the value of $\pi_{1,2}(x(t))$ by considering whether $d_{i,j}(x(t))$ is bigger or smaller than 0.

For example, if at time $t \geq 0$ $d_{1,2}(x(t)) > 0$, then $\pi_{1,2}(x(t)) = 1$ and if $d_{1,2}(x(t)) < 0$, then $\pi_{1,2}(x(t)) = 0$. This, together with the value of $z_{2,1}(t)$, immediately gives the value of $\Pi_{1,2}(x(t))$.

We need to use other fluid equations when at least one of the two pools is not full. If, for example $z_{1,1}(t) + z_{2,1}(t) < m_1(t)$, then necessarily $q_1(t) = 0 < k_{1,2}$, so that

$$\dot{z}_{1,2}(t) = -\mu_{1,2} z_{1,2}(t) \quad \text{and} \quad \dot{z}_{1,1}(t) = \lambda_1 - \mu_{1,1} z_{1,1}(t).$$

The evolution of $z_{2,1}$ in this case is determined by whether $q_2(t) < k_{2,1}$ or $q_2(t) \geq k_{2,1}$. In the first case $z_{2,1}(t)$ must be strictly decreasing at time t if it is positive, or remain at 0 otherwise. In the latter case, when $q_2(t) \geq k_{2,1}$, the excess fluid - that is not routed to pool 2 and does not abandon, if such excess fluid exists - is flowing to pool 1. We thus have

$$\dot{z}_{2,1}(t) = \begin{cases} -\mu_{2,1} z_{2,1}(t) & \text{if } q_2(t) < k_{2,1} \\ -\mu_{2,1} z_{2,1}(t) + (\lambda_2 - \mu_{2,2} z_{2,2}(t) - \mu_{1,2} z_{1,2}(t) - \theta_2 k_{2,1})^+ & \text{if } q_2(t) = k_{2,1} \end{cases} \tag{27}$$

Similar reasonings lead to the fluid model of $z_{1,2}$ when pool 1 is full, but pool 2 has spare capacity.

If both pools have spare capacity at time t, then $q_1(t) = q_2(t)$ and

$$\dot{z}_{i,j}(t) = -\mu_{i,j} z_{i,j}(t) \quad \text{and} \quad \dot{z}_{i,i}(t) = \lambda_i - \mu_{i,i} z_{i,i}(t), \quad i, j = 1, 2, \quad i \neq j.$$

To compute the solution x over an interval $[0, T]$ we employ the classical Euler method. Given a step size h and the time T, the number of iterations needed is $N \equiv T/h$. Let $\dot{x} = \Psi(x)$, where $\Psi(x)$ is the RHS of the appropriate ODE, e.g., if both pools are full, then $\Psi(x)$ is the RHS of (20). Given $x(0)$, we can compute $x(h)$ using the first Euler step: $x(h) = x(0) + h\Psi(x(0))$. Given $x(h)$ we can compute $\Pi_{1,2}(x(h))$ and $\Pi_{2,1}(x(h))$, if needed, and then compute $x(2h)$ using the second Euler step. In general, the solution to the ODE is computed via

$$x((k+1)h) = x(kh) + h\Psi(x(kh)), \quad 0 \leq k \leq N,$$

where at each step, if $x(kh) \in \mathbb{B}_{1,2}$ or $x(kh) \in \mathbb{B}_{2,1}$, we can compute $\Pi_{1,2}(kh)$ and $\Pi_{2,1}(kh)$ as explained above.

The algorithm just described remains unchanged when the ratio parameters are general (not equal to 1), except that the sets $\mathbb{A}_{i,j}$ and the computations of $\pi_{i,j}$ are more complicated (the FTSP's are no longer BD processes). We refer to [27] for these more complicated settings.

Remark 6.1. If at iteration $k \geq 0$ the solution lies outside the set $\mathbb{B}_{1,2} \cup \mathbb{B}_{2,1}$ then, due to the discreetness of the algorithm, there is a need to ensure that the boundary is not missed in the following iterations. Hence, if in the k^{th} iteration $d_{1,2}(x(kh)) > 0$ (< 0) and in the $(k+1)^{st}$ iteration $d_{1,2}((x(k+1)h)) < 0$ (> 0), then the boundary $d_{1,2}$ necessarily was missed, because the fluid is continuous, and so we set $d_{1,2}((x(k+1)h)) = 0$. We then check whether $x((k+1)h) \in \mathbb{A}_{1,2}((k+1)h)$, compute $\pi_{1,2}(x(k+1)h)$ and use its value to compute the value in the $(k+2)^{nd}$ iteration. It is significant that *we do not force the solution to be on the boundary,* e.g., we do not compute $q_1((k+1)h)$ and use its value to compute $q_2((k+1)h)$ via

$$q_2((k+1)h) = q_1((k+1)h) - k_{1,2}. \tag{28}$$

We solve the six-dimensional ODE in (20), and if indeed (28) holds whenever it should, then we have a good indication that the algorithm works. That is, we can check at which iteration the boundary $\mathbb{B}_{1,2}$ was hit, and then observe if $q_1(t) - q_2(t) = k_{1,2}$ over an interval for which we have indication that this should hold. (Of course, the solution to the algorithm might leave the boundary for legitimate reasons, i.e., because the fluid model leaves it.)

7 Numerical Examples

We now study three examples. The first two are piecewise continuous models, whereas the third is for a general time-varying model. In all three examples the system starts empty, so that we also check the numerical algorithm in periods when (19) does not hold, as in §5.4.

We compare the numerical solutions to the ODE to simulations, to see how well the fluid approximated stochastic systems. In the first two examples we simulate three systems, each can be considered as a component in a sequence $\{\bar{X}^n : n \geq 1\}$. In the smallest system we take 50 agents in each service pool, in the middle one there are 100 agents in a pool, and the largest has 400 agents in each pool, i.e., we simulate $\bar{X}^n$ for $n = 50, 100, 400$. That allows us to observe the "convergence" of the stochastic system to the fluid approximation. We plot the fluid and simulation results together, normalized to $n = 10$. (E.g., for the system with 400 agents in each pool we divide all processes by 40.)

The following parameters are used for all three simulations:

$\mu_{1,1} = \mu_{2,2} = 1$; $\mu_{1,2} = \mu_{2,1} = 0.8$, $\theta_1 = \theta_2 = 0.5$. In addition, we take $r_{1,2} = r_{2,1} = 1$. We take $k^n_{1,2} = k^n_{2,1} = 0.3n$; $\tau_{1,2} = \tau_{2,1} = 0.02n$, so that, for $n = 50, 100, 400$, we have $k^n_{1,2} = k^n_{2,1} = 15, 30, 120$ and $\tau_{1,2} = \tau_{2,1} = 1, 2, 8$, respectively.

7.1 A Single Overload Incident

The first example aims to check whether FQR-ART detects overloads automatically when they occur and starts sharing in the right direction, and whether, once an overload incident is over, FQR-ART avoids oscillations. In particular, over the time interval $[0, 60]$ the arrival rates are as follows: $\lambda^n_2 = n$ throughout that time interval. Over $[0, 20)$ and $[40, 60]$ the arrival rate to pool 1 is $\lambda^n_1 = n$. Hence, both pools are normally loaded during these two subintervals. However, during the interval $[20, 40)$ the arrival rate of class 1 changes to $\lambda^n_1 = 1.4n$, so that, during $[20, 40)$ the system is overloaded, and pool 2 should be helping class 1.

We compare the solution to the fluid equations, solved using the algorithm, to an average of 1000 independent simulation runs for the three cases $n = 50, 100, 400$. The results are shown in Figures 10-12 below. In addition Figure 13 plots $q_1 - r_{1,2}q_2 - k_{1,2}$. The fact that shortly after time 20 the value is 0, is a strong indication that the numerical solution is correct, because during most of the overload period, when sharing takes place, it should hold that $d_{1,2}(x(t)) = 0$.

The simulation experiments indicate that the fluid model approximates well the mean behavior of the

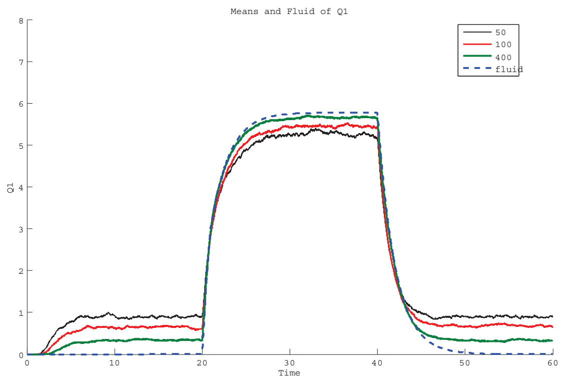


Figure 10: Fluid vs. simulations of Q_1^n

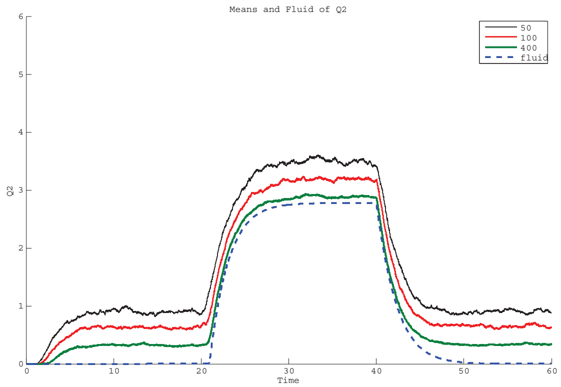


Figure 11: Fluid vs. simulations of Q_2^n

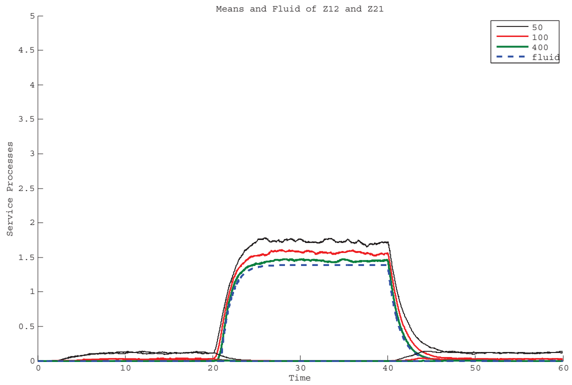


Figure 12: Fluid vs. simulations of $Z_{1,2}$

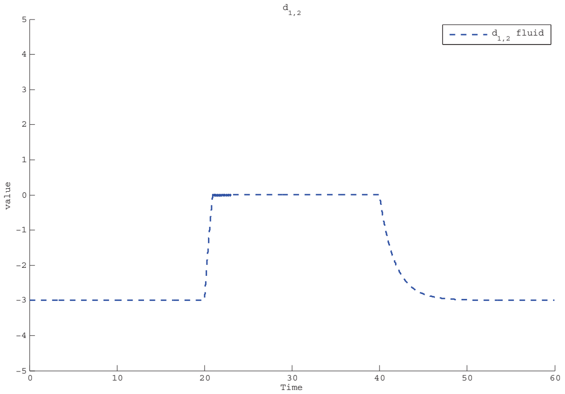


Figure 13: plot of $q_1 - r_{1,2}q_2 - k_{1,2}$

system even for relatively small systems, e.g., when $n = 50$. Of course, the accuracy of the approximation grows as n becomes larger. The simulation experiments show that FQR-ART quickly detects the overload and the correct direction of sharing. Moreover, the control ensures that there are no oscillations, as in §4.

Another observation is that when the system is normally loaded and there is no sharing, the fluid model, which has null queues, does not describe the queues well. In those cases there is an increased importance to stochastic refinements for the queues. If there is only negligible sharing, as FQR-ART ensures, then such stochastic refinements are well approximated by diffusion limits for the Erlang A model, as in [14].

7.2 Switching Overloads

In the second example we consider an overloaded system, with pool 1 being overloaded initially, and with the direction of overload switching after some time, making pool 2 overloaded. Specifically, we let the arrival rates be $\lambda_1^n = 1.4n$ and $\lambda_2^n = n$ over $[0, 20)$, and $\lambda_1^n = n$, $\lambda_2^n = 1.4n$ on $[20, 40]$. The results are plotted in Figures 14-16.

Figure 17 plots $q_1 - r_{1,2}q_2 - k_{1,2}$ and $r_{2,1}q_2 - q_1 - k_{2,1}$. Once again, the fact that the appropriate difference process equals to 0 shortly after the corresponding overload begins is an indication that the solution to the ODE is correct, since each queue is calculated via the AP, without forcing the relations $d_{1,2}(x(t)) = 0$ and

$d_{2,1}(x(t)) = 0$.

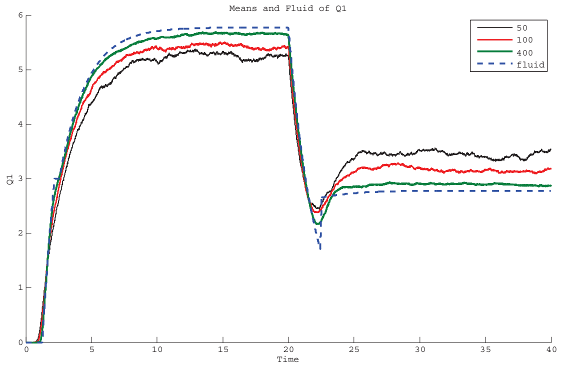


Figure 14: Fluid vs. simulations of Q_1^n

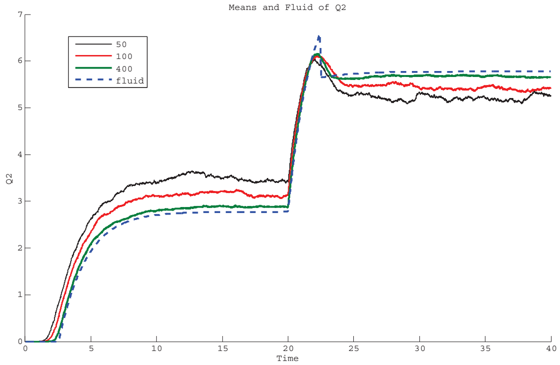


Figure 15: Fluid vs. simulations of Q_2^n

As in the figures in §7.1, it is easily seen from the figures above that the fluid model approaches a fixed point, so long as the arrival rates are fixed. Then, once a change in the rates occurs, the fluid goes through a new transient period until it relaxes in a new fixed point. We elaborate in §8 below.

7.3 General Non-stationary Model with switching overloads

We next test our algorithm in challenging example. That example is unrealistic, since the arrival rates and staffing functions are not likely to change as drastically as in that example (at least in call centers). We assume that the arrival rate to pool 1 over the time period $[0, 20)$ is sinusoidal. We further assume that management anticipated the basic sinusoidal pattern of the arrival rate, but did not anticipated the magnitude, so that pool 1 is overloaded. To accommodate the sinusoidal pattern, we assume that staffing follows the appropriate *infinite-server* approximation; see, e.g., Equation (9) in [11]. The purpose of that staffing rule in our setting, is to stabilize the system at a fixed point eventually, as in the examples above. In particular, for $t \in [20, 40]$ we take

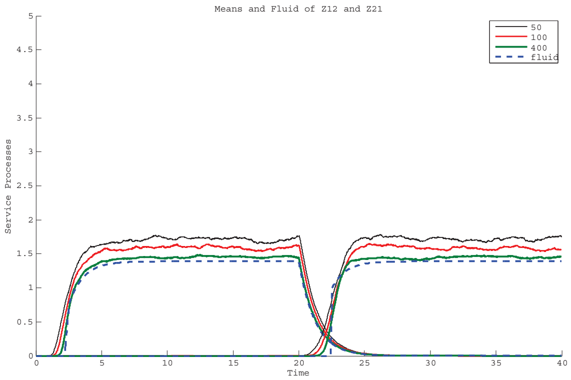


Figure 16: Fluid vs. simulations of $Z_{1,2}^n$ and $Z_{2,1}^n$

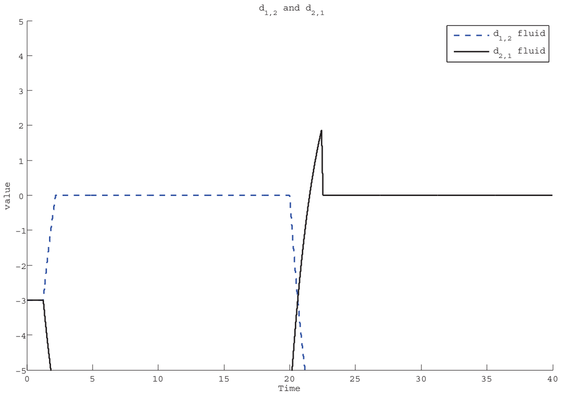


Figure 17: Fluid difference processes

$\lambda_1^n(t) = 1.3n + 0.1n\sin(t)$ and $m_1^n(t) = n + 0.05n[\sin(t) - \cos(t)]$; $\lambda_2^n(t) = n$ and $m_2^n(t) = n$.

Then, on the time interval $[20, 40]$ the overload switches, with pool 2 becoming overloaded and experiencing a sinusoidal arrival rate. However, we now take fixed staffing in both service pools. In particular, the parameters over the second overload interval $[20, 40]$ is

$\lambda_1^n(t) = n$ and $m_1^n(t) = n$; $\lambda_2^n(t) = 1.1n + 0.1n\sin(t)$ and $m_2^n(t) = n$.

Thus, we test two overload settings in this example. In the first interval, we can see whether the fluid approximation stabilizes. Since there is sharing of class-1 customers, previous results in the literature (see the review in §1.2) do not extend directly to our case. In the second interval, we expect to see a sinusoidal behavior of the system, because the staffing in both pools is fixed. In particular, the fluid model should not approach a fixed point after the switch at time $t = 20$.

We now simulate the case $n = 100$ and $n = 400$ to make the figures clear, since in the second period. Figures 18–21 demonstrate the effectiveness of the fluid model and the numerical algorithm. First, Figure 18 shows that the control stabilizes the two queues at the appropriate relation, and that the algorithm captures that well. (We emphasize again that each queue is computed independently via the AP.) As expected, the fluid over $[0, 20)$ approaches a fixed point, and exhibits a sinusoidal behavior after $t = 20$, with the accuracy of the approximation growing together with the size of the system.

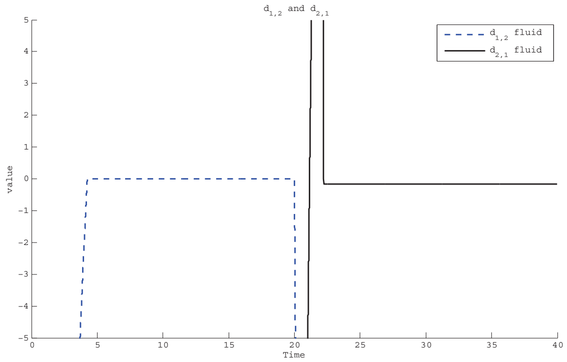


Figure 18: Fluid vs. simulation, $d_{1,2}$ and $d_{2,1}$

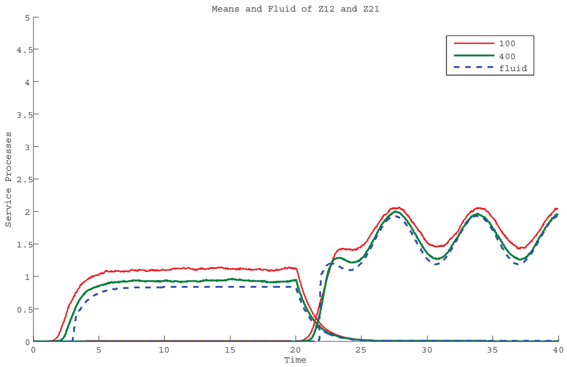


Figure 19: Fluid vs. simulations of $Z_{1,2}^n$ and $Z_{2,1}^n$

Removing Agents When the Staffing function is Decreasing. In applications, there is a need to determine how to remove agents from the pool when the staffing function is decreasing, when there are not enough idle agents and there is a need to remove busy agents in order for the actual number of agents to be close to the staffing function. (Note that, since the staffing functions are assumed to be continuous almost everywhere, we cannot have the actual staffing be exactly equal to the staffing function, since the number of agents in the stochastic system is an integer and changes by ± 1.) In a call-center, it is unlikely that agents be removed while still serving customers. Hence, in our simulations an agent is removed from the pool only if he is idle, or immediately after he completes a service. Specifically, at each service completion we check wether there are "too many" agents in the pool (compared to the ceiling of of the staffing function: $\lceil m_1^n(t) \rceil$), and only then the newly-available agent is removed from the pool, if needed. Figure 22 shows

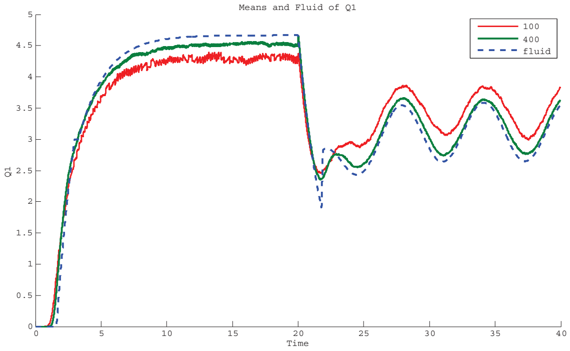


Figure 20: Fluid vs. simulations of Q_1^n

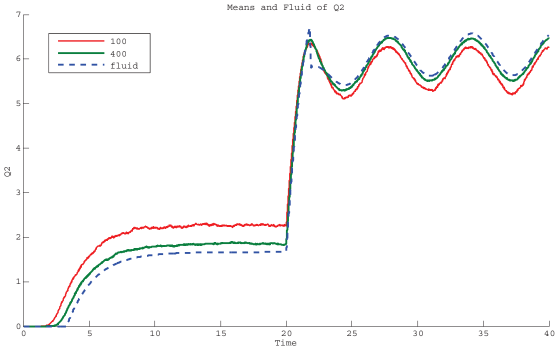


Figure 21: Fluid vs. simulations of Q_2^n

the actual number of agents in Pool 1 for the case $n = 100$ (the average of the 1000 simulations), and the staffing function $m_1^n(t)$ given above. Clearly, the staffing rule just explained provides an actual staffing that is very close the target one. Moreover, at time $t = 20$ the number of agents has a downward jump, and we see that the staffing rule we employed mimics this jump, although, unlike in the fluid approximation, the actual staffing can only jump down by 1 at a time. This behavior is to be expected, since there are many service completions over short time intervals in large systems.

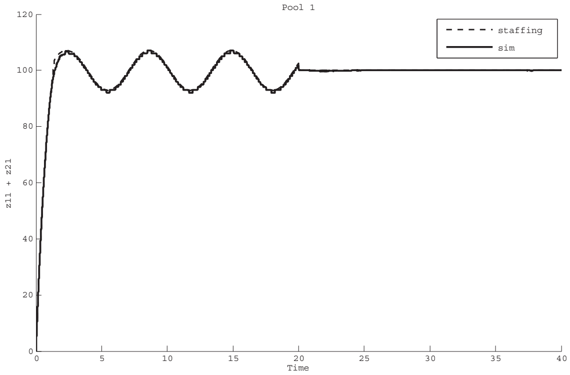


Figure 22: Fluid vs. simulations: Number of agents in pool 1

8 Stationarity and Quasistationarity

The fluid model is said to be stationary if it is fixed at a single point $x^* \in \mathbb{R}_6$, and x^* is said to be a *stationary point*. In other words, x^* is a stationary point if $x(s) = x^*$ for some $s \geq 0$, implies that $x(t) = x^*$ for all $t \geq s$ or, equivalently, $\dot{x}(t) = 0$, for all $t \geq s$. Hence, a stationary point is also a fixed point of the fluid model.

For each fixed arrival rates and staffing function there exists a unique stationary point. This was proved in §8 of [27] for the case in which class 1 is overloaded and one-way sharing is performed with pool 1 helping. The exact same reasonings can be used to prove this statement when class 2 is overloaded with one way sharing. If both classes are normally loaded or underloaded and there is no sharing in the fluid approximation, then this claim follows easily from known results for the Erlang A model.

The models we consider here are nonstationary. However, as long as the arrival rates and staffing functions remain fixed, or if the staffing changes appropriately with respect to the arrival pattern, as in Period 1 of the example in §7.3, then there exists a point that would have been a stationary point, if the rates of the system would never change. We call such points *quasistationary* points.

In the stationary model considered in our previous papers, we have proved that the unique stationary point of the fluid model is exponentially stable, i.e., that the fluid model converges to that points exponentially fast, for any initial condition $x(0)$ which has sharing in the appropriate direction; see Theorem 9.2 in [27]. It stands to reason that a similar result holds for any quasistationary point, provided they exist but, unlike in [27], here this should hold for **any initial condition**. We do not prove this assertion, but observe that the simulation experiments in §§7.1 and 7.2, as well as in the first period of the example in §7.3, suggest that this claim holds, at least in many reasonable examples.

9 Conclusions and Further Research

In this paper we studied a time-varying X model experiencing periods of overloads. We observed that our previous control, FQR-T, which was designed to respond to unexpected overloads, needs to be adjusted to the time-varying environment, in order to respond quickly to changed in the direction of the overload and then prevent behavior that FQR-T may cause in these more complex settings. We thus suggested using a modification of the control, which we named FQR-ART. With FQR-ART, the one-way sharing rule is weakened by adding the release thresholds. To avoid oscillations of the service process, which in turn causes congestion collapse, we further replaced the activation thresholds of FQR-T, that were suggested to be as in (5), with thresholds that are of order n, i.e., satisfying (6). We then developed a fluid model, based on a stochastic AP, to approximate the transient evolution of the time-varying system under FQR-ART, together with an algorithm to numerically compute that fluid model. Simulation experiments suggest that this fluid model captures the main dynamics of the system, even in extreme cases, as the one considered in (7.3).

Future Research. As the literature review show, oscillatory behavior and congested collapse are important phenomenons in other stochastic networks. Our methods and insights here could potentially have impact in other applications as well. However, it remains to show that FQR-ART indeed prevents the bad behavior from occurring. Specifically, it remains to prove that the ODE in (20) posses a unique solution and that, at least under some reasonable regularity conditions, oscillatory behavior is prevented.

Due to the importance of piecewise constant models, it will be useful to prove that for each of the intervals of fixed rates and staffing, and *regardless of the starting point of the fluid model at any such interval*, there is convergence to the appropriate quasistationary point.

While we view the above research questions to be the most fundamental from the mathematical point of view, it is also beneficial to prove that the fluid model arises as the many-server heavy-traffic fluid limit of a properly-scaled sequence of X systems operating under FQR-ART. (Note that such a proof will also imply that the ODE (20) has a unique solution.)

References

[1] Aksin, Z., Armony, M., Mehrotra, V., 2007. The modern call center: a multi-disciplinary perspective on operations management research. *Production Oper. Management*, 16(6) 655-688.

[2] Armony, M., Israelit, S., Mandelbaum, A., Marmor, Y., Tseytlin, Y., Yom-tov, G., 2010. Patient Flow in Hospitals: A Data-Based Queueing-Science Perspective. *Working paper*

[3] Boyle, A., Beniuk, K., Higginson, I., Atkinson, P., 2012. Emergency department crowding: Time for interventions and policy evaluations. *Emergency Medicine International*.

[4] Choudhury G. L., Leung, K. K., Whitt, W., 1995. Efficiently Providing Multiple Grades of Service with Protection Against Overloads in Shared Resources. *AT&T Technical Journal*, **74** (4), 50–63.

[5] Chan, C. W., Armony, M, Bambos, N., 2011. Fairness in overloaded parallel queues. *Working paper*

[6] Chan, C. W., Yom-Tov, G., Escobar, G., 2011. When to use Speedup: An Examination of Intensive Care Units with Readmissions. *Working paper*

[7] J. G. Dai, 1995. On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability*, **5** (1), 49–77.

[8] Deo, S., Gurvich, I., 2011. Centralized vs. Decentralized Ambulance Diversion: A Network Perspective. *Mangement Sci.* **57** (7), 1300–1319.

[9] Doshi, B., Heffes, H., 1986. Overload performance of several processor queueing disciplines for the M/M/1 queue. *IEEE Transactions on Communications*, **34** (6), 538–546.

[10] Erramilli, A., Forys, L.J., 1991. Oscillations and chaos in a flow model of a switching system. *IEEE journal on Selected Areas in Communications*, **9** (2), 171–178.

[11] Feldman, Z., Mandelbaum, A., Massey, W. A., Whitt, W. 2008. Staffing of time-varying queues to achieve time-stable performance. *Management Sci.* **54** (2), 324–338.

[12] Forys, L.J. and Im, C.S., Henderson, W., 1988. Analysis of Load Box Testing for Voice Switches. *ITC-12, Torino*.

[13] Gans, N., Koole, G., Mandelbaum, A., 2003. Telephone call centers: tutorial, review and research prospects. *Manufacturing and Service Opns. Mgmt.*, 5, 79–141.

[14] Garnett, O., Mandelbaum, A., Reiman, M. I., 2002. Designing a call center with impatient customers. *Manufacturing Service Oper. Management*, **4** (3), 208–227.

[15] Gurvich, I., Perry, O., 2012. Overflow networks: Approximations and implications to call-center outsourcing. *Operations Research*, **60** (4), 996–1009.

[16] Harrison J. M., Zeevi., A., 2005. A method for staffing large call centers using stochastic fluid models. *Manufacturing Service Oper. Management*, **7** (1), 20–36.

[17] Jennings, O.B., Mandelbaum, A., Massey, W.A., Whitt, W., 1996. Server staffing to meet time-varying demand. *Mangement Sci.* **42** (10), 1383–1394.

[18] Khalil, H. K., 2002. *Nonlinear Systems.* Prentice Hall, New Jersey.

[19] Klemm, F. Le Boudec, J.Y., Aberer, K., 2006. Congestion control for distributed hash tables. In: Fifth IEEE International Symposium on Network Computing and Applications.

[20] Körner U., 1991. Overload control of SPC systems. *Teletraffic and data traffic*

[21] Liu, Y., Whitt, W., 2011. Large-Time Asymptotics for the $G_t/M_t/s_t + GI_t$ Many-Server Fluid Queue with Abandonment. *Queueing Syst*, **67**, 145–182.

[22] Liu, Y., Whitt, W., 2012a. Stabilizing Customer Abandonment in Many-Server Queues with Time-Varying Arrivals. *Operations Research*, forthcoming.

[23] Liu, Y., Whitt, W., 2012b. A Many-Server Fluid Limit for the $G_t/GI/s_t + GI$ Queueing Model Experiencing Periods of Overloading. *Operations Research Letters*, forthcoming.

[24] Pang, G., Talreja, R., Whitt, W., 2007. Martingale proofs of many-server heavy-traffic limits for Markovian queues. *Probability Surveys*, **4**, 193–267.

[25] Perry, O., Whitt, W., 2009. Responding to unexpected overloads in large-scale service systems. *Management Sci.*, **55** (8), 1353–1367.

[26] Perry, O., Whitt, W., 2011a. A fluid approximation for service systems responding to unexpected overloads. *Operations Res.*, **59** (5), 1159–1170. Available at: http://www.columbia.edu/∼ww2040/allpapers.html

[27] Perry, O., Whitt, W., 2011b. An ODE for an overloaded X model involving a stochastic averaging principle. *Stochastic Systems*, **1** (1), 17–66.

[28] Perry, O., Whitt, W., 2012. A fluid limit for an overloaded X model via an averaging principle. *Mathematics of Operations Research*, forthcoming. Available at: http://www.columbia.edu/∼ww2040/allpapers.html

[29] Perry, O., Whitt W., 2012. Diffusion approximation for an overloaded X model via a stochastic averaging principle. *Working paper*. Available at: http://www.columbia.edu/∼ww2040/allpapers.html

[30] Powell, E.S., Khare, R.K., Venkatesh, A.K., Van Roo, B.D., Adams, J.G., Reinhardt, G., 2012. The relationship between inpatient discharge timing and emergency department boarding. *The Journal of emergency medicine*, **42** (2), 186–196.

[31] Schulzrinne, H., Kurose, J.F., Towsley, D., 1990. Congestion control for real-time traffic in high-speed networks. IEEE proceeding in Ninth Annual Joint Conference of the IEEE Computer and Communication Societies, 543–550.

[32] Shah, D., Wischik, D., 2011. Fluid models of congestion collapse in overloaded switched networks. *Queueing Syst*, 69, 121–143.

[33] Shi, P., Chou, M., Dai, J. G., Ding, D., Sim, J., 2012. Hospital Inpatient Operations: Mathematical Models and Managerial Insights. *Working paper*

[34] Stolyar, A. L., Tezcan, T., 2010. Contorl of systems with flexible multi-server pools: a shadow routing approach. Queueing syst **66** 1–51.

[35] Stolyar, A. L., Tezcan, T., 2011. Shadow-routing based control of flexible multiserver pools in overload. *Operations Research* **59** (6), 1427–1444.

[36] Teschl, G., 2009. *Ordinary Differential Equations and Dynamical Systems*, Universität Wien. Available online: www.mat.univie.ac.at/∼gerald/ftp/book-ode/ode.pdf

[37] Whitt, W., 2002. *Stochastic-Process Limits*, New York, Springer, 2002.

[38] Whitt, W., 2004. Efficiency-driven heavy-traffic approximations for many-server queues with abandonments. *Management Sci.* **50** (10) 1449–1461.

[39] Whitt, W., 2006. Fluid Models for Multiserver Queues with Abandonments. *Operations Research*, **54** (1) 37–54.

A Choosing the Activation Thresholds

For an example, we consider the case of recovering from an overload period with pool-1 helping queue 2. We assume that $z_{2,1}(0) > 0$ but $\lambda_i \leq \mu_{i,i} m_i$, $i = 1, 2$. If $\lambda_1 = \mu_{1,1} m_1$, then the condition $z_{2,1}(0) > 0$ implies that pool 1 is overloaded, since $z_{2,1}(t) > 0$ for all $t \geq 0$. The maximal service rate in pool 1 over time is

$$\mu_{1,1}(m_1 - z_{1,2}(t)) + \mu_{2,1} z_{2,1}(t) = \lambda_1 - (\mu_{1,1} - \mu_{2,1}) z_{2,1}(t) < \lambda_1, \quad t \geq 0,$$

where we assume that type-1 agents are less efficient serving class-2 customers, i.e., that $\mu_{2,1} < \mu_{1,1}$. Our goal is to choose the thresholds on the queues and the release thresholds such that $q_1(T) < k_{1,2}$, for $T \equiv \inf\{t \geq 0 : z_{2,1}(t) = \tau_{2,1}\}$. Since the arrival rate to pool 2 is such that that pool is not overloaded, queue 2 quickly decreases after the arrival rates have changed back to normal, so that no more class-2 customers are sent to pool 1. Hence, $z_{2,1}$ satisfies the ODE $\dot{z}_{2,1}(t) = -\mu_{2,1} z_{2,1}(t)$, $t \geq 0$, whose unique solution, given the initial condition, is

$$z_{2,1}(t) = z_{2,1}(0) e^{-\mu_{2,1} t}, \quad t \in [0, T], \tag{29}$$

Recalling that $\lambda_1 = \mu_{1,1} m_1$, the fluid model of $q_1(t)$ over $t \in [0, T]$ satisfies the ODE

$$\begin{aligned} \dot{q}_1(t) &= \lambda_1 - \mu_{1,1}(m_1 - z_{2,1}(t)) - \mu_{2,1} z_{2,1}(t) - \theta_1 q_1(t) \\ &= (\mu_{1,1} - \mu_{2,1}) z_{2,1}(t) - \theta_1 q_1(t), \quad t \in [0, T]. \end{aligned} \tag{30}$$

It follows from (29) and (30) that

$$\frac{d}{dt}(e^{\theta_1 t}q_1(t)) = (\mu_{1,1} - \mu_{2,1})\int_0^t z_{2,1}(0)e^{-\mu_2 s}ds$$

so that, for $t \in [0, T]$,

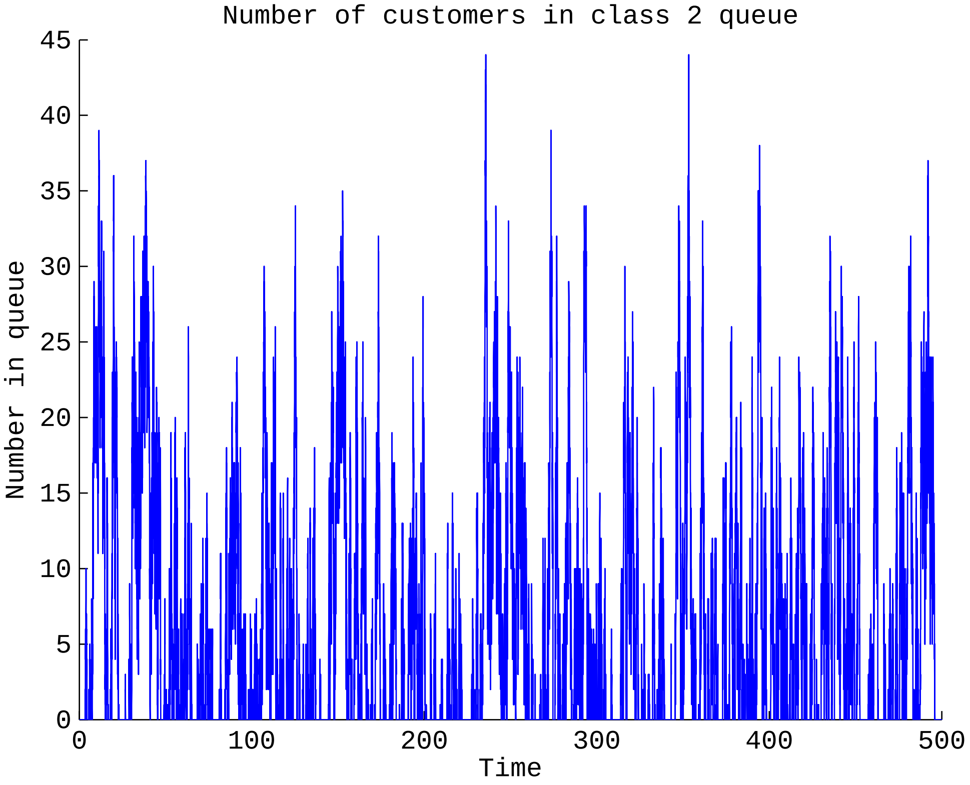
$$q_1(t) = \frac{(\mu_{2,1} - \mu_{1,1})z_{2,1}(0)}{\mu_{2,1}}e^{-(\mu_{2,1}+\theta_1)t} + \left(q_1(0) - \frac{(\mu_{2,1} - \mu_{1,1})z_{2,1}(0)}{\mu_{2,1}}\right)e^{-\theta_1 t}. \tag{31}$$

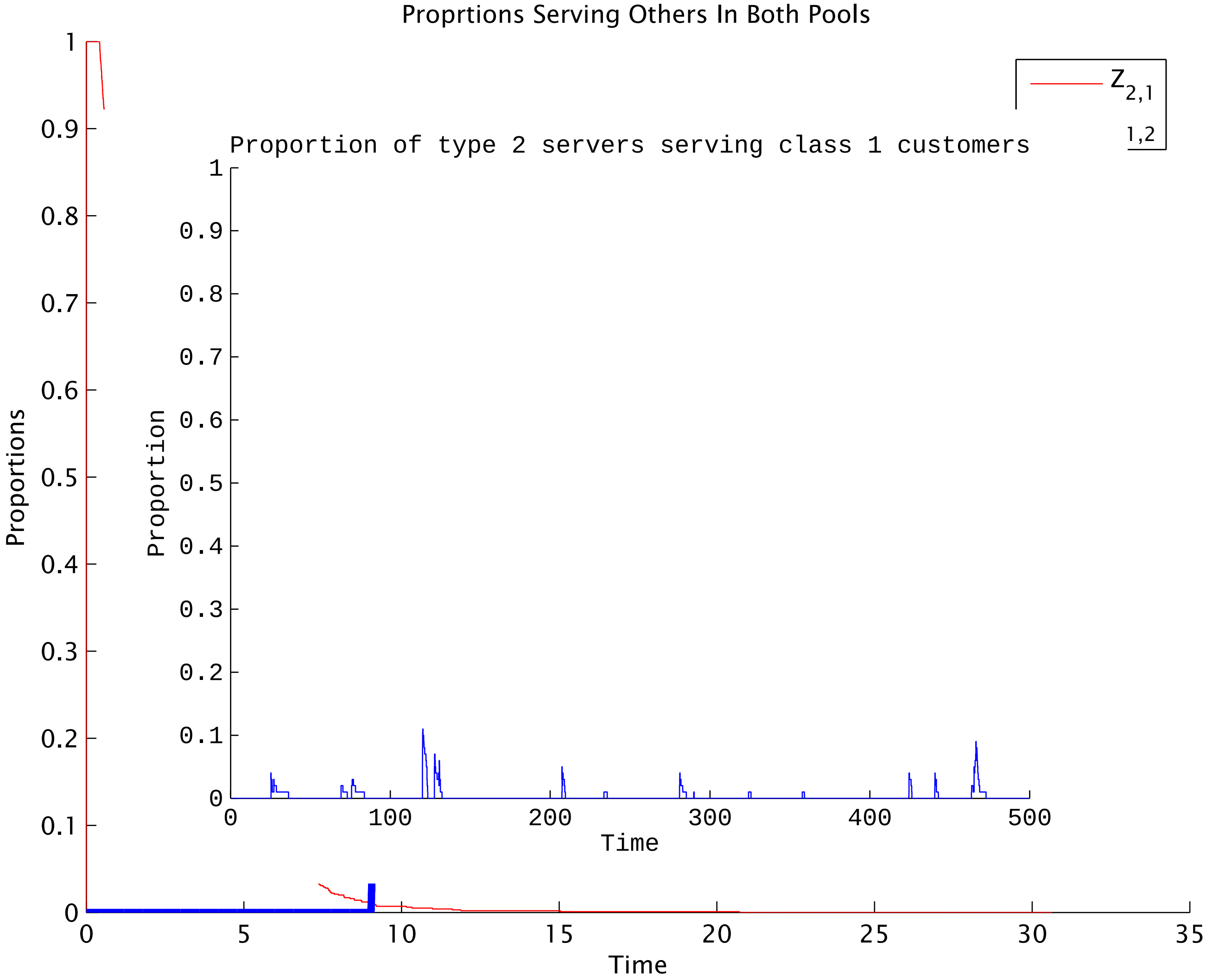
Since we cannot know a-priori what values $q_1(0)$ and $z_{2,1}(0)$ will have, we use the upper bounds λ_1/θ_1 and m_1, respectively. The bound for $q_1(0)$ is achieved by considering all the type-1 servers working with class 2, so that none of the class-1 customers receives any help, and the only output from the queue is due to abandonment.

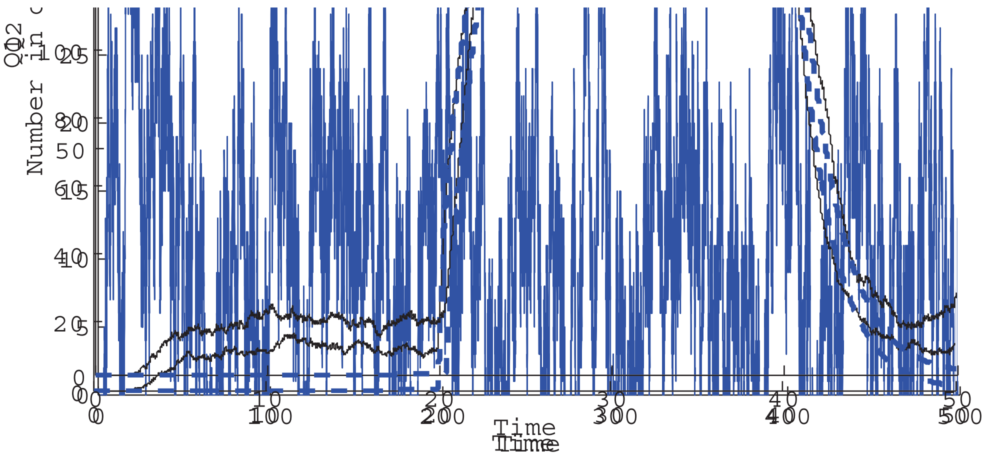
Taking our example in the beginning of the section with $z_{2,1}(0) \leq m_1 < 1$ and $q_1(0) \leq \lambda_1/\theta_1 = 2$,

$$\tau_{2,1} = z_{2,1}(T) \leq e^{-0.5T} \quad \text{and} \quad q_1(T) \leq e^T + e^{-T/2}.$$

We first choose $\tau_{2,1}$ to be $0.01m_1$ (if our system has 100 agents in pool 1, then we might choose the release threshold to be a bit larger, e.g., $0.03m_1$). Then by time $T = 9.21$ the threshold must have been crossed (in the fluid limit). Plugging $T = 9.21$ in (31) gives $q_1(9.21) \leq 1.01$. We thus take $k_{2,1}^n > 0.066n$. For example, for $n = 1000$, we should have $k_{2,1}^n > 101$. However, taking into consideration stochastic fluctuations, which are of order $\sqrt{n}$ in normally loaded systems (operating in the QED regime) we see that $k_{2,1} = 101$ may be too small. Recall that we considered $k_{2,1}^n = 150$ as an appropriate threshold for the purpose of preventing sharing when the two pools are normally loaded, and for detecting unexpected overloads. The above analysis shows that this value for $k_{2,1}^n$ is appropriate also from fluid considerations.


Number of customers in class 2 queue
Number in queue
Time
0
5
10
15
20
25
30
35
40
45
0
100
200
300
400
500


Proprtions Serving Others In Both Pools
Proportion of type 2 servers serving class 1 customers
Z2,1
1,2
Proportions
Proportion
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
Time
0
5
10
15
20
25
30
35
Time


Q12
Number in c
100
80
60
50
40
20
25
20
15
10
5
0
0
0
100
200
300
400
500
Time
Time